Distributed Optimization using Virtual and Real Game Strategies for Aerodynamic Design

Zhili Tang, Jean-Antoine Desideri, Jacques Périaux

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Distributed Optimization using Virtual and Real Game Strategies for Aerodynamic Design

Zhili Tang — Jean-Antoine Désidéri — Jacques Périaux

N° 4543

September 2002

THÈME 4

ISSN 0249-6399 ISRN INRIA/RR--4543--FR+ENG

INRIA
SOPHIA ANTIPOLIS

Distributed Optimization using Virtual and Real Game Strategies for Aerodynamic Design

Zhili Tang , Jean-Antoine Désidéri , Jacques Périaux

Thème 4 — Simulation et optimisation
de systèmes complexes
Projet OPALE

Rapport de recherche n° 4543 — September 2002 — 53 pages

Abstract: This report approaches the question of multi-disciplinary optimization for optimum shape design in Aerodynamics using game theory. The employed optimizer is based on control theory, which produces very robust optimization algorithms particularly well suited for problems of constrained optimizations. We perform an airfoil drag reduction under the constraint of constant C_l, and a lift maximization under the constraint of constant C_d. Here, we introduce a scalar adjoint variable to satisfy these constraints. Comparing with the unconstrained case, these constraints can be easily implemented by introducing an additional scalar algebraic adjoint equation. Furthermore, the above methodological ingredients are combined with a formulation derived from *Game Theory* to treat multi-point airfoil optimization. Airfoil shapes are optimized according to various aerodynamic criteria (under conflict). Each 'player' in a symmetric Nash game optimizes one's own criterion using information provided by the others. The Nash equilibrium then corresponds to the solution of a multi-point optimization. Subsonic/Transonic flows around lifting airfoils are analyzed by *Eulerian* computations. Several kinds of airfoil splittings and aerodynamic design cases are considered illustrating virtual and real game strategies. Successful design results confirm the validity and efficency of the present design method in a parallel computing environment.

Key-words: Distributed Optimization, Distributed Parallel Computing, Multi-objective Optimization, Optimal Control Theory, Euler Equations, Adjoint Equations, Nash Game, Virtual & Real Game strategies, Shape parameterization

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles, BP 93, 06902 Sophia Antipolis Cedex (France)
Téléphone : +33 4 92 38 77 77 —Télécopie : +33 4 92 38 77 65

Optimisation distribuée utilisant des stratégies de jeux réels ou virtuels en conception optimale de forme aérodynamique

Résumé : Ce rapport approche la question de l'optimisation multidisciplinaire en conception optimale de forme aérodynamique par une formulation de type théorie des jeux. L'optimiseur est construit par les outils classiques de la théorie du contrôle, ce qui conduit à des algorithmes d'optimisation très robustes particulièrement bien adaptés aux problèmes contraints. On réalise la réduction de traînée d'un profil, à portance constante, et la maximisation de sa portance, à traînée constante. Ces contraintes sont traitées par l'introduction d'une variable adjointe scalaire supplémentaire. En comparaison au cas de l'optimisation sans contraintes, cette généralisation n'introduit qu'une nouvelle équation adjointe scalaire. De plus, ces ingrédients méthodologiques sont combinés à une formulation dérivée de la théorie des jeux pour traiter le problème d'optimisation multipoint d'un profil d'aile. Les formes sont optimisées vis à vis de différents critères aérodynamiques (en conflit). Chaque joueur, dans un jeu symétrique de Nash, optimise un critère donné en tenant compte d'informations fournies par les autres. L'équilibre de Nash ainsi atteint correspond à la solution d'un problème d'optimisation multipoint. Des écoulements subsoniques et transsoniques autour de profils portants sont analysés par un code eulérien. Plusieurs types de partage des territoires associés aux joueurs et plusieurs cas-tests de conception aérodynamique sont considérés illustrant diverses stratégies de jeux réels ou virtuels. Les résultats confirment la validité et l'efficacité de la méthode pour la conception dans un environnement de calcul parallèle.

Mots-clés : Optimisation distribuée, calcul parallèle distribué, optimisation multiobjectif, théorie du contrôle optimal, équations d'Euler, équations adjointes, jeu de Nash, stratégies de jeu réel ou virtuel, paramétrisation de forme

Contents

1 Introduction

Multi-disciplinary optimization (MDO) is gaining importance in Aeronautics in order to meet with society's needs. MDO can be described as a methodology for the design of systems where the interaction between several disciplines must be considered, and where the designer is free to significantly affect the system performance in more than one discipline. In the literature, contributions to single-discipline and/or single-point design optimization abound. The two main challenges of MDO are computational expense and organization complexity. The ingredients of classical approaches in automated optimization include shape parameterization, calculus of variations and control theory, and descent solution methods[1][2].

Among the many techniques that have been developed, the inverse design method[2][3][4] is perhaps the most widely used. This method is severely restrictive because its depends on the experience and knowledge of the designer to establish desirable velocity or pressure distributions. In addition, the method does not lend itself to the imposition of constraints[5].

The next most popular approach is what G. Kuruvula et al.[5] referred to as the 'loosely coupled optimization (LCO)'. In this approach, the analysis problem is solved many times to find the gradient of the cost function with finite difference formula. This gradient is used by a 'black box' optimizer to find the best change to the design variables. The computation cost of the brute-force method usually exorbitant due to the fact that n+1 aerodynamic analysis are necessary, n being the number of parameters defining the geometry to optimize. Examples of this approach are fund in Refs[6][7][8]. The LCO method can be improved by analytically evaluating the sensitivity derivatives that are needed to update the design variables[8]. Usually, this requires the inversion of an extremely large matrix. For three-dimensional problems, the size of this matrix can render the method impractical with current computer technology[5].

Genetic algorithms(GAs)[9][10] offer an alternative to gradient-guided search. The GAs algorithm treats a set of design points in the design space as a population of individuals that produces another set of points as a generation of parents produces the next generation of children. A GA comprises a mechanism for pairing up the design points into the pair of parents for transfer of the parent characteristic to children and for mutations that occasionally endow children with features absent in either parent. The mechanism favors probabilitical creation of children that are better than parents in terms of the objective function and satisfaction of constraints. The mutation mechanism in GAs is particularly important to prevent the process from ending up in a local minimum[9]. Although GAs don't require gradient information, they are also highly cost in term of CPU time.

In the past several years, a large focus[11][12][13][14][15][16] came back again on design algorithms in the area of gradient-based optimization. Research has focused on several methods for obtaining sensitivity information, and many of these approaches rely on adjoint-variable formulation for efficiently computing sensitivity derivatives. The adjoint variable technique is particularly attractive for aerodynamic design problem in which a large number of design variables are considered, yet relatively few constraints. Pironneau had earlier initiated studies of the use of control theory for optimum shape design of systems governed by elliptic equation[23][24]. Táasan, Kuruvila and Salas have proposed a one-shot approch[5]

in which the constraint represented by the flow equations need only be satisfied by the final converged design solutions. Adjoint methods have also been used by Baysal[13][15][18] and Eleshaky[25], and by Cabuk and Modi[26], and by Anderson[16].

In this report, a method based on optimal control theory[11][12], derived from the continuous Euler equations, to evaluate the gradient is presented. This method, which G. Kuruvula et al.[5] referred to as 'tightly coupled optimization(TCO)', requires the solution of adjoint problem, in addition to the analysis problem, to evaluate the gradient of cost function. The complexity of the adjoint problem is equivalent to that of the analysis problem, therefore the computational cost of this method is approximatively twice the evaluation of the aerodynamic function, regardless the number of design parameters, and is well suited to shape design problems. As we mentioned at the end of last paragraph, only relatively few constraints can be implemented in this method, but optimization with constraints is practical in aerodynamic design, such as to minimize the drag with constraint of constant lift. Recent publications [17][18] present the constrained aerodynamic shape optimization using the adjoint-variable technique, in which the lift constraint is satisfied by adjusting the incidence angle α based on Cl_α, the flow solver is run in a fixed lift mode, and the objective function is also modified correspondingly. There are two main drawbacks in this method: (1) the cost function becomes complex, where Cl_α is included, (2) the incidence angle has to be modified during the iterations of time integration of state equations. In this report, the lift constraint is dealt with as an explicit constraint in the optimizer; here, we introduce a scalar adjoint variable to achieve this. According the control theory, this yields an additional adjoint equation for the Cl derivatives, but fortunately, this adjoint equation is a scalar linear algebraic equation involving the points around the airfoil surface, and it can be solved easily.

The application of the adjoint approach has produced remarkable improvements in the ability to design for certain types of optimal behavior, but it is not possible in general to treat arbitrary cost function, especially in multidisciplinary design. Usually, there are two different ways to solve a multi-disciplinary optimization problem: the first one is to make a linear combination of the different criteria with different weights and to minimize the resulting function, but this kind of linear combination does not exist in general when the different criteria are in conflict, moreover the weights assigned to some criterion may not be suitable and the resulting function may lack significance, the second one is to use the Pareto concept[19] to form a set of solutions that are strictly better than the remaining ones in the whole search space when considering all the objectives simultaneously, the drawback of this method is the high cost in CPU time. The goal of this paper is to introduce a novel approach combining Adjoint-variable technique with a formulation derived from Game Theory to treat multi-point airfoil optimization problems. Transonic flows about lifting airfoils are analyzed by eulerian computations[20]. Airfoils shapes are optimized according to various aerodynamic criteria. In a symmetric Nash game, the notion of 'players' is introduced and each player attempts to optimize one's own target with exchange of symmetric information with the others[21]. A Nash equilibrium is reached when each player, constrained by the strategy of the others, cannot improve further one's own target. Specific virtual and real symmetric

Nash games are implemented to set up an optimization strategy for design under conflict. Here, the Adjoint-variable technique provides a cheap optimizer, and Nash Game contributes its good feature in distributed computing and gives an easy way to achieve the equillibrium between the criteria under conflict in a multi-disciplinary design.

2 General Formulation of The Adjoint Approach to Optimal Design

Before embarking on a detailed derivation of the adjoint formulation for optimal design using the Euler equation, it is helpful to summarize the general abstract description of the adjoint approach which has been thoroughly documented in references[11][12].

The progress of the design procedure is measured in terms of a cost function I, which could be, for example the drag coefficient or the lift to drag ratio. For flow about an airfoil or wing, the aerodynamic properties which define the cost function are functions of the flow-field variables (w) and the physical location of the boundary, which may be represented by the function b, say, Then

$$I = I(w, b) \tag{1}$$

and a change in b results in a change

$$\delta I = \left[\frac{\partial I^T}{\partial w}\right]_I \delta w + \left[\frac{\partial I^T}{\partial b}\right]_{II} \delta b \tag{2}$$

in the cost function. Here, the subscripts I and II are used to distinguish the contributions due to the variation δw in the flow solution from the change associated directly with the modification δb in the shape. This notation is introduced to assist in grouping the numerous terms that arise during the derivation of the Euler adjoint operator, so that it remains feasible to recognize the basic structure of the approach as it sketched in the present section.

Using control theory, the governing equations of the flow field are introduced as a constraint in such a way that the final expression for the gradient does not require multiple flow solution. This corresponds to eliminating δw from (2).

Suppose that the governing equation R which expresses the dependence of w and b within the flow-field domain D can be written as

$$R(w, b) = 0 \tag{3}$$

Then δw is determined from the equation

$$\delta R = \left[\frac{\partial R}{\partial w}\right]_I \delta w + \left[\frac{\partial R}{\partial b}\right]_{II} \delta b = 0 \tag{4}$$

Next, introducing a Lagrange Multiplier Ψ, we have

$$\begin{aligned}\delta I &= \frac{\partial I^T}{\partial w}\delta w + \frac{\partial I^T}{\partial b}\delta b - \Psi^T\left(\left[\frac{\partial R}{\partial w}\right]\delta w + \left[\frac{\partial R}{\partial b}\right]\delta b\right) \\ &= \left\{\frac{\partial I^T}{\partial w} - \Psi^T\left[\frac{\partial R}{\partial w}\right]\right\}_I \delta w + \left\{\frac{\partial I^T}{\partial b} - \Psi^T\left[\frac{\partial R}{\partial b}\right]\right\}_{II} \delta b\end{aligned} \tag{5}$$

Choosing Ψ to satisfy the adjoint equation

$$\left[\frac{\partial R}{\partial w}\right]^T \Psi = \frac{\partial I}{\partial w} \tag{6}$$

the first term is eliminated, and we find that

$$\delta I = \dot{g}\delta b \tag{7}$$

where

$$\dot{g} = \frac{\partial I^T}{\partial b} - \Psi^T\left[\frac{\partial R}{\partial b}\right] \tag{8}$$

The advantage is that (7) is independent of δw, with the result that the gradient of I with respect to an arbitary number of design variables can be determined without the need for additional flow-field evaluations. In the case that (3) is a partial differential equation, the adjoint equation (6) is also a partial differential equation and the determination of the appropriate boundary conditions requires careful mathematical treatment.

The computational cost of a single design cycle is roughly equivalent to the cost of two flow solutions since the adjoint problem has similiar complexity. When the number of design variables becomes large, the computational efficiency of the control theory approach over traditional approach, which requires direct evaluation of gradients by individually varying each design variable and recomputing the flow field, becomes compelling.

Once equation (8) is established, an improvement can be made with a shape change

$$\delta b = -\lambda\dot{g} \tag{9}$$

where λ is positive, and small enough so that the first variation is an accurate estimate of δI. The variation in the cost function then becomes

$$\delta I = -\lambda\dot{g}^T\dot{g} \leq 0 \tag{10}$$

After making such a modification, the gradient can be recalculated and the process repeated to follow a path of steepest descent until a minimum is reached. In order to avoid violating constraints, such as a minimum acceptable wing thickness, the gradient may be projected into an allowable subspace within which the constraints are satisfied. In this way, a procedure is devised which necessarily converges at least to a local minimum.

3 Unconstrained Optimal Control Theory

In this section we consider the two-dimensional airfoil unconstrained optimization using the optimal control theory, the flow field is governed by Euler equations. Assume that the flow field domain is **D** defined by the airfoil shape c as the internal boundary and the remote closed curve **B** as the far-field boundary, e.g. figure 1.

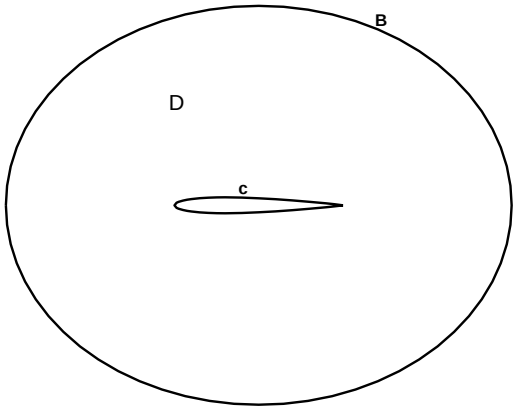

Figure 1: Domain **D** with design surface c and outer boundary **B**

Let's assume that p, ρ, u, v, e, h are the pressure, density, velocity components in x and y directions, total energy and total enthalpy per unit volume respectively, Then the two-dimensional Euler equations in Cartesian coordinates are:

$$\frac{\partial w}{\partial t} + \frac{\partial f}{\partial x} + \frac{\partial g}{\partial y} = 0 \tag{11}$$

In the case of a perfect gas we have

$$p = (\gamma - 1)\left[e - \frac{1}{2}\rho(u^2 + v^2)\right] \qquad h = e + p \tag{12}$$

The vector w contains the conservative variables, f and g are the convective flux vectors in x and y directions respectively[20] defined as follows,

$$w = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ e \end{pmatrix}, f = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (e+p)u \end{pmatrix}, g = \begin{pmatrix} \rho v \\ \rho vu \\ \rho v^2 + p \\ (e+p)v \end{pmatrix} \tag{13}$$

In order to treat the solid boundary condition precisely, we introduce the curvilinear coordinates, and generate a structured mesh around the airfoil using conformal mapping techniques, figure 2 below shows this kind of mesh around a single airfoil. The two-dimensional Euler equations in curvilinear coordinates have the following form:

$$\frac{\partial W}{\partial t} + \frac{\partial F}{\partial \xi} + \frac{\partial G}{\partial \eta} = 0 \tag{14}$$

where $W = J^{-1}w$, F and G are convective flux in ξ and η direction respectively, $F = \xi_x/Jf + \xi_y/Jg$, $G = \eta_x/Jf + \eta_y/Jg$, J is the Jacobian of the transformation.

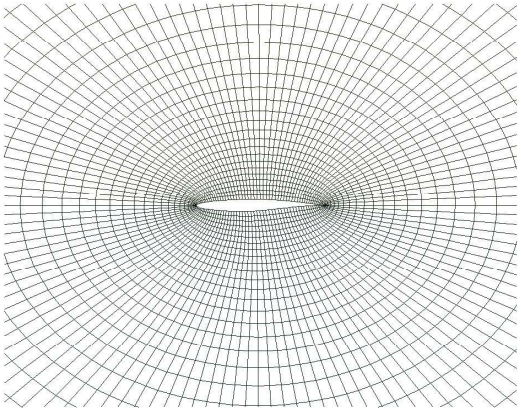

Figure 2: Partial view of **O**-type mesh

For the flow around the airfoil or wing, the aerodynamics characteristics give the value of the cost function. In general, the cost function is defined as follows in the aerodynamic optimal design problem:

$$I = \oint_c \phi ds = \oint_c \phi \frac{ds}{d\xi} d\xi \tag{15}$$

which is a function of the flow variables and the boundary of body as we mentioned in section 2, where ϕ is a function of the pressure p, or the pressure difference between the actual pressure and a target pressure in inverse design problem, or the pressure drag coefficient of a transonic or supersonic airfoil in drag reduction problem. A variation in the airfoil shape produces a variation in the cost function

$$\delta I = \oint_c \phi_p \delta p \frac{ds}{d\xi} d\xi + \oint_c \phi \frac{d\delta s}{d\xi} d\xi \tag{16}$$

which is a function of the variation in the flow field variables and the variation of airfoil shape. As it has already been mentioned by Jameson[11], the other terms can be easily determinied except δp which would require the greater computing time if we were using the traditional method of finite differences, because then, the flow would have to be solved as many times as the number of design variables at every iteration. However, by using control theory, the flow equations are introduced as constraint in such a way that we only need to solve the flow equations once every iteration.

At first, we define some Jacobian matrices for convenience in the following derivations:

$$A_1 = \frac{\partial f}{\partial w}, A_2 = \frac{\partial g}{\partial w}, C_1 = \frac{\partial F}{\partial w} = \frac{\xi_x}{J}A_1 + \frac{\xi_y}{J}A_2, C_2 = \frac{\partial G}{\partial w} = \frac{\eta_x}{J}A_1 + \frac{\eta_y}{J}A_2 \tag{17}$$

Here we consider the steady state flow, and the variation of the Euler equations are:

$$\frac{\partial}{\partial \xi}(\delta F) + \frac{\partial}{\partial \eta}(\delta G) = 0 \tag{18}$$

as the result of $\delta f = \frac{\partial f}{\partial w} = A_1 \delta w$, $\delta g = \frac{\partial g}{\partial w} = A_2 \delta w$, hence

$$\delta F = C_1 \delta w + \delta(\frac{\xi_x}{J}) f + \delta(\frac{\xi_y}{J}) g \qquad \delta G = C_2 \delta w + \delta(\frac{\eta_x}{J}) f + \delta(\frac{\eta_y}{J}) g \tag{19}$$

As we mentioned previously, now introduce the Lagrangian multiplier Ψ, and integrate the variational form of the Euler equations within the domain D:

$$\int_D \Psi^T \left[\frac{\partial}{\partial \xi}(\delta F) + \frac{\partial}{\partial \eta}(\delta G) \right] d\xi d\eta = 0 \tag{20}$$

The left hand side of the above equation can be divided into two parts. As we use the O-type grid in this paper, it is periodic in ξ direction, therefore the first integral part of equation (20) is

$$\int_D \Psi^T \frac{\partial \delta F}{\partial \xi} d\xi d\eta = -\int_D \frac{\partial \Psi^T}{\partial \xi} \delta F d\xi d\eta \tag{21}$$

and the second integral part of equation (20)

$$\int_D \Psi^T \frac{\partial \delta G}{\partial \eta} d\xi d\eta = \oint_B \Psi^T \delta G d\xi - \oint_c \Psi^T \delta G d\xi - \int_D \frac{\partial \Psi^T}{\partial \eta} \delta G d\xi d\eta \tag{22}$$

If the far field boundary B is maintained during the design, then $\delta G = 0$ for inflow wave, that is to say $\Psi^T \delta G = 0$. if we can make $\Psi^T \delta G = 0$ for the outflow wave, the above integration can be written:

$$\int_D \Psi^T \frac{\partial \delta G}{\partial \eta} d\xi d\eta = -\oint_c \Psi^T \delta G d\xi - \int_D \frac{\partial \Psi^T}{\partial \eta} \delta G d\xi d\eta \tag{23}$$

and equation (20) becomes

$$\int_D \left(\frac{\partial \Psi^T}{\partial \xi} \delta F + \frac{\partial \Psi^T}{\partial \eta} \delta G \right) d\xi d\eta = -\oint_c \Psi^T \delta G d\xi \tag{24}$$

Consequently, the variation of cost function can be written:

$$\delta I = \oint_c \phi_p \delta p \frac{ds}{d\xi} d\xi + \oint_c \phi \frac{d\delta s}{d\xi} d\xi + \int_D \left(\frac{\partial \Psi^T}{\partial \xi} \delta F + \frac{\partial \Psi^T}{\partial \eta} \delta G \right) d\xi d\eta + \oint_c \Psi^T \delta G d\xi \tag{25}$$

Now rearrange the third and fourth terms in the right hand side of above equation, we have

$$\begin{aligned}\int_D \left(\frac{\partial \Psi^T}{\partial \xi}\delta F + \frac{\partial \Psi^T}{\partial \eta}\delta G\right) d\xi d\eta = \int_D \left[\frac{\partial \Psi^T}{\partial \xi}C_1 + \frac{\partial \Psi^T}{\partial \eta}C_2\right] \delta w d\xi d\eta + \\ \int_D \left\{\frac{\partial \Psi^T}{\partial \xi}\left[\delta(\frac{\xi_x}{J})f + \delta(\frac{\xi_y}{J})g\right] + \frac{\partial \Psi^T}{\partial \eta}\left[\delta(\frac{\eta_x}{J})f + \delta(\frac{\eta_y}{J})g\right]\right\} d\xi d\eta\end{aligned} \tag{26}$$

On the airfoil surface c, according the flow field boundary condition $V = 0$, we have $\delta G|_c = \delta p \left[\begin{array}{cccc} 0 & \frac{\eta_x}{J} & \frac{\eta_y}{J} & 0 \end{array}\right]^T + p \left[\begin{array}{cccc} 0 & \delta(\frac{\eta_x}{J}) & (\delta\frac{\eta_y}{J}) & 0 \end{array}\right]^T$, so the fourth term becomes

$$\oint_c \Psi^T \delta G d\xi = \oint_c \left(\Psi_2 \frac{\eta_x}{J} + \Psi_3 \frac{\eta_y}{J}\right) \delta p d\xi + \oint_c \left[\Psi_2 \delta\left(\frac{\eta_x}{J}\right) + \Psi_3 \delta\left(\frac{\eta_y}{J}\right)\right] p d\xi \tag{27}$$

Introduce (16) (17) into (15), to get:

$$\begin{aligned}\delta I = \oint_c \phi_p \delta p \frac{ds}{d\xi} d\xi + \oint_c \phi \frac{d\delta s}{d\xi} d\xi + \int_D \left[\frac{\partial \Psi^T}{\partial \xi}C_1 + \frac{\partial \Psi^T}{\partial \eta}C_2\right] \delta w d\xi d\eta + \\ \int_D \left\{\frac{\partial \Psi^T}{\partial \xi}\left[\delta(\frac{\xi_x}{J})f + \delta(\frac{\xi_y}{J})g\right] + \frac{\partial \Psi^T}{\partial \eta}\left[\delta(\frac{\eta_x}{J})f + \delta(\frac{\eta_y}{J})g\right]\right\} d\xi d\eta + \\ \oint_c (\Psi_2 \frac{\eta_x}{J} + \Psi_3 \frac{\eta_y}{J}) \delta p d\xi + \oint_c \left[\Psi_2 \delta(\frac{\eta_x}{J}) + \Psi_3 \delta(\frac{\eta_y}{J})\right] p d\xi\end{aligned} \tag{28}$$

To eliminate the terms which include flow field variation, we choose Ψ to satisfy the following adjoint equation

$$\left\{\begin{array}{ll} \dfrac{\partial \Psi}{\partial t} - C_1^T \dfrac{\partial \Psi}{\partial \xi} - C_2^T \dfrac{\partial \Psi}{\partial \eta} = 0 & in\ \ D \\ \Psi_2 \dfrac{\eta_x}{J} + \Psi_3 \dfrac{\eta_y}{J} = -\phi_p \dfrac{ds}{d\xi} & on\ \ c \\ \Psi^T \delta G = 0 & on\ \ B \end{array}\right. \tag{29}$$

As a result, the variation of cost function simplifies to

$$\begin{aligned}\delta I = \oint_c \phi \frac{d\delta s}{d\xi} d\xi + \oint_c \left[\Psi_2 \delta(\frac{\eta_x}{J}) + \Psi_3 \delta(\frac{\eta_y}{J})\right] p d\xi + \\ \int_D \left\{\frac{\partial \Psi^T}{\partial \xi}\left[\delta(\frac{\xi_x}{J})f + \delta(\frac{\xi_y}{J})g\right] + \frac{\partial \Psi^T}{\partial \eta}\left[\delta(\frac{\eta_x}{J})f + \delta(\frac{\eta_y}{J})g\right]\right\} d\xi d\eta\end{aligned} \tag{30}$$

In this expression, as we expected, the flow variations have disappeared and the variation of the cost function can be computed in terms of the metric variations flow field variables and adjoint variables, so that equation (30) can be evaluated without requiring any additional computation of the flow field. The new problem is only to solve the adjoint equations (29), that, in general, has a complexity similar to that of the flow. To integrate the third term at right hand side of above equation (30) by parts, and let $\bar{f}$ and $\bar{g}$ are vectors of convective flux vectors f and g without pressure, then equation (30) can be further written as

$$\begin{aligned}\delta I = -\int_D \Psi^T \left\{ \frac{\partial}{\partial \xi}\left[\delta(\frac{\xi_x}{J})f + \delta(\frac{\xi_y}{J})g\right] + \frac{\partial}{\partial \eta}\left[\delta(\frac{\eta_x}{J})f + \delta(\frac{\eta_y}{J})g\right]\right\} d\xi d\eta \\ - \oint_c \Psi^T \left[\delta(\frac{\eta_x}{J})\bar{f} + \delta(\frac{\eta_y}{J})\bar{g}\right] d\xi + \oint_c \phi \frac{d\delta s}{d\xi} d\xi\end{aligned} \tag{31}$$

concerning the integration area in the first term at right hand side of above equation, we find that $\frac{\partial}{\partial \xi}\left[\delta(\frac{\xi_x}{J})f + \delta(\frac{\xi_y}{J})g\right] + \frac{\partial}{\partial \eta}\left[\delta(\frac{\eta_x}{J})f + \delta(\frac{\eta_y}{J})g\right]$ has the similiar form with the form of steady state Euler equation (14) under the curvilinear coordinates, the difference is that metric terms in Equation (14) are replaced by their variations. For convenience, we assume

$$\frac{\partial}{\partial \xi}\left[\delta(\frac{\xi_x}{J})f + \delta(\frac{\xi_y}{J})g\right] + \frac{\partial}{\partial \eta}\left[\delta(\frac{\eta_x}{J})f + \delta(\frac{\eta_y}{J})g\right] = \delta R'_{es} \tag{32}$$

above equation can be calculated easily by using a similiar process as the computation of residuals of the Euler equations. This simplifies the computation compared with equation (30); furthermore, we write

$$\delta(\frac{\eta_x}{J})\bar{f} + \delta(\frac{\eta_y}{J})\bar{g} = \delta \bar{G}' \tag{33}$$

hence, equation (31) can be written as

$$\delta I = -\int_D \Psi^T \delta R'_{es} d\xi d\eta - \oint_c \Psi^T \delta \bar{G}' d\xi + \oint_c \phi \frac{d\delta s}{d\xi} d\xi \tag{34}$$

Assume that the airfoil shape is represented by a series of smooth functions, that is to say

$$f(x) = \sum_{i=1}^{i=N} b_i f_i(x) \tag{35}$$

where $f(x)$ is the curve of airfoil contour, $f_i(x)$ is shape function, b_i is design variables, N is number of design variables. The gradient of the cost function is defined as

$$\dot{g}(b_i) = \frac{\delta I}{\delta b_i} \tag{36}$$

Once the gradient is established, the same process, described in (9) (10), can be used to determine a modification on the airfoil shape.

4 Constrained Optimal Control Theory

In a practical aerodynamic design context, a design usually has to satisfy a number of constraints. A constraint is said to be active if it is either an equality constraint or an inequality constraint, the number of active constraints should not exceed the total number of design parameters, otherwise the problem is over-constrained and will not have a solution in general. The computational design system will then attempt to find the optimal solution to the problem, subject to the constraints, using the most appropriate optimization technique. There are many modifications to the standard unconstrained optimizations methods to handle constraints, often through the use of Lagrange multipliers. Here we treat two kinds of constrained optimizations, one is the drag minimization problem under the constraint of constant lift in transonic regime, the other is the lift maximization problem under the constraints of constant drag in subsonic regime. In this report all of these constraints are treated as equality constraints. That is to say, for the drag minimization problem, we try to design a new optimal airfoil which has the same lift coefficient as the initial one but the drag is reduced in transonic flow; for the lift maximizition problem, we try to design a new optimal airfoil whose drag is not greater than or at most equal to the initial value but lift is increased in subsonic flow. In this report, we follow the concept of control theory to treat the constraint explicitly in the optimizer by introducing a scalar adjoint variable; additionally this yields a linear algebraic adjoint equation which is easy to solve.

4.1 Drag Minimization Under the Constraint of Constant Lift

Drag minimization is a very practical problem in aerodynamic engineering. In general, the lift is also reduced while reducing the drag because drag is very closely connected with lift. Some authors solve this problem by difining the design problem as the maximization of lift to drag ratio C_l/C_d; as we know, C_l and C_d are simultaneously reduced during the design even though the ratio is increased. In practice, the lift must always be kept as the same value as the aircraft weight during the cruise. So we treat the drag reduction problem as the following constrained optimization:

$$\begin{cases} minI = \oint_c \phi ds \\ C_l = constant \end{cases} \tag{37}$$

Here, ϕ includes the drag coefficient, we keep the free stream Mach number M_∞ and incidence angle α unchanged during the design. For the pure drag reduction problem, we can define $I = Cd$; in practical design, we only want to make a small modification to the original airfoil and try to reduce its drag under the fixed lift mode; in this case the cost function is defined in general as $I = \frac{1}{2}\Omega_1 \oint_c (p-p_d)^2 ds + \Omega_2 Cd$. In case the Mach number and the angle of attack are maintained during the optimization, the lift and drag coefficients of an airfoil

are

$$C_l = C_y \cos(\alpha) - C_x \sin(\alpha) \qquad C_d = C_x \cos(\alpha) + C_y \sin(\alpha)$$
$$C_x = \frac{2}{\gamma M_\infty^2 \bar{C}} \oint_c \frac{p}{p_\infty} dy \qquad C_y = -\frac{2}{\gamma M_\infty^2 \bar{C}} \oint_c \frac{p}{p_\infty} dx \tag{38}$$

where, C_x and C_y are the coefficients of force in x and y directions respectively, $\bar{c}$ is the airfoil reference chord length . The computational coordinate system is as follows

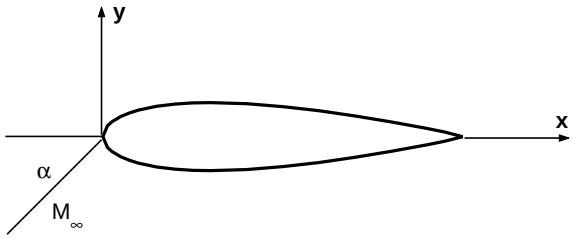

Figure 3: Sketch of computational coordinates

In this report, the nondimensionalization of pressure is $\bar{p} = p/p_\infty$, $\bar{p}$ is dimensionaless pressure, for the convenience, we write $\bar{p}$ as p in the following derivations. As we know, $\delta M_\infty = 0$ and $\delta\alpha = 0$, so $\delta C_l = \delta C_y \cos(\alpha) - \delta C_x \sin(\alpha) = 0$, it is well known that for a small angle of attack C_x and $\sin(\alpha)$ are smal compared with C_y and $\cos(\alpha)$, so that the variation of lift coefficient can be simplified as

$$\delta C_l \approx \delta C_y \cos(\alpha) = 0 \Rightarrow \delta C_y \approx 0 \tag{39}$$

hence

$$\delta C_d = \delta C_x \cos(\alpha) + \delta C_y \sin(\alpha) \approx \delta C_x \cos(\alpha) \tag{40}$$

The above derivation tells us that the drag minimization problem considered here can be simplified as follows:

$$\begin{cases} minI = \dfrac{\Omega_1}{2} \oint_c (p - p_d)^2 ds + \Omega_2 C_d \\ \qquad subject\ \ to\ \ \delta C_y \cos(\alpha) = 0 \end{cases} \tag{41}$$

Following the same derivation process as in section 3, to evaluate the variation in the cost function, we have

$$\delta I = \Omega_1 \oint_c (p - p_d)\delta p \frac{ds}{d\xi} d\xi + \frac{\Omega_1}{2} \oint_c (p - p_d)^2 \frac{d\delta s}{d\xi} d\xi + \frac{2\Omega_2 \cos(\alpha)}{\gamma M_\infty^2 \bar{C}} \oint_c \left(\delta p \frac{dy}{d\xi} + p \frac{d\delta y}{d\xi} \right) d\xi \tag{42}$$

and the variation in the lift constraint is:

$$\delta C_l \approx \delta C_y \cos(\alpha) = \frac{2\cos(\alpha)}{\gamma M_\infty^2 \bar{C}} \oint_c \left(-\delta p \frac{dx}{d\xi} - p \frac{d\delta x}{d\xi} \right) d\xi = 0 \tag{43}$$

Now, we introduce a scalar Lagrange multiplier Θ, such that

$$\frac{2\cos(\alpha)\Theta}{\gamma M_\infty^2 \bar{C}} \oint_c \left(-\delta p \frac{dx}{d\xi} - p \frac{d\delta x}{d\xi} \right) d\xi = 0 \tag{44}$$

In control theory, equation (20) and (44) are considered as constraints in the design system, so that the variation in the cost function becomes:

$$\begin{gathered} \delta I = \Omega_1 \oint_c (p - p_d) \delta p \frac{ds}{d\xi} d\xi + \frac{\Omega_1}{2} \oint_c (p - p_d)^2 \frac{d\delta s}{d\xi} d\xi + \\ \frac{2\Omega_2 \cos(\alpha)}{\gamma M_\infty^2 \bar{C}} \oint_c \left(\delta p \frac{dy}{d\xi} + p \frac{d\delta y}{d\xi} \right) d\xi \\ + \frac{2\cos(\alpha)\Theta}{\gamma M_\infty^2 \bar{C}} \oint_c \left(-\delta p \frac{dx}{d\xi} - p \frac{d\delta x}{d\xi} \right) d\xi - \int_D \Psi^T \left[\frac{\partial}{\partial \xi}(\delta F) + \frac{\partial}{\partial \eta}(\delta G) \right] d\xi d\eta \end{gathered} \tag{45}$$

Furthermore, this can be rearranged as

$$\begin{gathered} \delta I = \Omega_1 \oint_c (p - p_d) \delta p \frac{ds}{d\xi} d\xi + \frac{\Omega_1}{2} \oint_c (p - p_d)^2 \frac{d\delta s}{d\xi} d\xi + \\ \frac{2\cos(\alpha)}{\gamma M_\infty^2 \bar{C}} \oint_c \delta p \left(\Omega_2 \frac{dy}{d\xi} - \Theta \frac{dx}{d\xi} \right) d\xi + \frac{2\cos(\alpha)}{\gamma M_\infty^2 \bar{C}} \oint_c p \left(\Omega_2 \frac{d\delta y}{d\xi} - \Theta \frac{d\delta x}{d\xi} \right) d\xi + \\ \int_D \left[\frac{\partial \Psi^T}{\partial \xi} C_1 + \frac{\partial \Psi^T}{\partial \eta} C_2 \right] \delta w d\xi d\eta + \\ \int_D \left\{ \frac{\partial \Psi^T}{\partial \xi} \left[\delta(\frac{\xi_x}{J}) f + \delta(\frac{\xi_y}{J}) g \right] + \frac{\partial \Psi^T}{\partial \eta} \left[\delta(\frac{\eta_x}{J}) f + \delta(\frac{\eta_y}{J}) g \right] \right\} d\xi d\eta + \\ \oint_c (\Psi_2 \frac{\eta_x}{J} + \Psi_3 \frac{\eta_y}{J}) \delta p d\xi + \oint_c \left[\Psi_2 \delta(\frac{\eta_x}{J}) + \Psi_3 \delta(\frac{\eta_y}{J}) \right] p d\xi \end{gathered} \tag{46}$$

Control theory requires that the variation of flow field variables should disappear to eliminate the additional redundent flow field analysis, so we have

$$\begin{cases} \dfrac{\partial \Psi}{\partial t} - C_1^T \dfrac{\partial \Psi}{\partial \xi} - C_2^T \dfrac{\partial \Psi}{\partial \eta} = 0 & in \;\; D \\ \Psi_2 \dfrac{\eta_x}{J} + \Psi_3 \dfrac{\eta_y}{J} = -\Omega_1 (p - p_d) - (1 - \sigma) \dfrac{2\Omega_2 \cos(\alpha)}{\gamma M_\infty^2 \bar{C}} \dfrac{dy}{d\xi} & on \;\; c \\ \Psi^T \delta G = 0 & on \;\; B \end{cases} \tag{47}$$

where $\sigma \in [0,1]$; in this work, we have set $\sigma = 0.4$, and Θ satisfies the following relationship

$$\Theta = \sigma\Omega_2 \frac{dy}{dx} \tag{48}$$

This is another adjoint equation arising from the lift constraint. Therefore, the variation in cost function can be simplified as

$$\begin{aligned}\delta I = \frac{\Omega_1}{2}\oint_c (p-p_d)^2 \frac{d\delta s}{d\xi}d\xi + \frac{2\cos(\alpha)}{\gamma M_\infty^2 \bar{C}}\oint_c p\left(\Omega_2\frac{d\delta y}{d\xi} - \Theta\frac{d\delta x}{d\xi}\right)d\xi + \\ \int_D \left\{\frac{\partial \Psi^T}{\partial \xi}\left[\delta(\frac{\xi_x}{J})f + \delta(\frac{\xi_y}{J})g\right] + \frac{\partial \Psi^T}{\partial \eta}\left[\delta(\frac{\eta_x}{J})f + \delta(\frac{\eta_y}{J})g\right]\right\}d\xi d\eta + \\ \oint_c \left[\Psi_2\delta(\frac{\eta_x}{J}) + \Psi_3\delta(\frac{\eta_y}{J})\right]p d\xi\end{aligned} \tag{49}$$

according to a similiar derivation as in section (3), above equation can be rewritten as:

$$\begin{aligned}\delta I = \frac{\Omega_1}{2}\oint_c (p-p_d)^2 \frac{d\delta s}{d\xi}d\xi + \frac{2\cos(\alpha)}{\gamma M_\infty^2 \bar{C}}\oint_c p\left(\Omega_2\frac{d\delta y}{d\xi} - \Theta\frac{d\delta x}{d\xi}\right)d\xi \\ - \int_D \Psi^T\left\{\frac{\partial}{\partial \xi}\left[\delta(\frac{\xi_x}{J})f + \delta(\frac{\xi_y}{J})g\right] + \frac{\partial}{\partial \eta}\left[\delta(\frac{\eta_x}{J})f + \delta(\frac{\eta_y}{J})g\right]\right\}d\xi d\eta \\ - \oint_c \Psi^T\left[\delta(\frac{\eta_x}{J})\bar{f} + \delta(\frac{\eta_y}{J})\bar{g}\right]d\xi\end{aligned} \tag{50}$$

Similiarly, we can further simplify to equation (50) using (32) and (33), so that the above formulation can be rewritten similiarly to (34), i.e.

$$\begin{aligned}\delta I = \frac{\Omega_1}{2}\oint_c (p-p_d)^2 \frac{d\delta s}{d\xi}d\xi + \frac{2\cos(\alpha)}{\gamma M_\infty^2 \bar{C}}\oint_c p\left(\Omega_2\frac{d\delta y}{d\xi} - \Theta\frac{d\delta x}{d\xi}\right)d\xi \\ - \int_D \psi^T \delta R_{es}^{'} d\xi d\eta - \oint_c \Psi^T \delta \bar{G}^{'} d\xi\end{aligned} \tag{51}$$

Once the airfoil shape is parameterized, we can calculate the gradient of the cost function with respect to design variables and modify the airfoil by (9) and (10), then repeat the process until convergence.

4.2 Lift Maximization Under the Constraint of Constant Drag

Usually, the lift maximization is not as important as the drag minimization problem because it can be easily realized by an increase in the angle of attack if current incidence angle is

far less than the stall angle, but the main drawback is that the drag is also increased while lift increased, and the pure lift maximization is impractical. Therefore we consider the lift maximization problem as a constrained optimization, that is to say, maximize lift under the constraint of constant drag using control theory and Euler equations, the optimization problem considered here is defined as:

$$\begin{cases} maxI = C_l \\ C_d = constant \end{cases} \tag{52}$$

where C_l and C_d are lift and drag coefficients defined in equation (38). As required by control theory, take the variation in the cost function, to get:

$$\begin{aligned} \delta I = & \frac{2}{\gamma M_\infty^2 \bar{C}} \oint_c \left(-\delta p \cos(\alpha) \frac{dx}{d\xi} - \delta p \sin(\alpha) \frac{dy}{d\xi} \right) d\xi \\ & + \frac{2}{\gamma M_\infty^2 \bar{C}} \oint_c \left(-p \cos(\alpha) \frac{d\delta x}{d\xi} - p \sin(\alpha) \frac{d\delta y}{d\xi} \right) d\xi \end{aligned} \tag{53}$$

Similiarly, take the variation in the drag constraint,

$$\begin{aligned} \delta C_d = & \frac{2}{\gamma M_\infty^2 \bar{C}} \oint_c \left(\delta p \cos(\alpha) \frac{dy}{d\xi} - \delta p \sin(\alpha) \frac{dx}{d\xi} \right) d\xi \\ & + \frac{2}{\gamma M_\infty^2 \bar{C}} \oint_c \left(p \cos(\alpha) \frac{d\delta y}{d\xi} - p \sin(\alpha) \frac{d\delta x}{d\xi} \right) d\xi = 0 \end{aligned} \tag{54}$$

This constraint should be satisfied during the design, so we introduce a scalar Lagrange multiplier Θ, such that

$$\begin{aligned} & \frac{2\Theta}{\gamma M_\infty^2 \bar{C}} \oint_c \left(\delta p \cos(\alpha) \frac{dy}{d\xi} - \delta p \sin(\alpha) \frac{dx}{d\xi} \right) d\xi \\ & + \frac{2\Theta}{\gamma M_\infty^2 \bar{C}} \oint_c \left(p \cos(\alpha) \frac{d\delta y}{d\xi} - p \sin(\alpha) \frac{d\delta x}{d\xi} \right) d\xi = 0 \end{aligned} \tag{55}$$

As we know, this and equation (20) will be used as the constraints to the system. Follow the same derivations as in section 3, to eliminate the terms including the variation in the

flow field variables, and get the following adjoint equation

$$\begin{cases} \dfrac{\partial \Psi}{\partial t} - C_1^T \dfrac{\partial \Psi}{\partial \xi} - C_2^T \dfrac{\partial \Psi}{\partial \eta} = 0 & in \;\; D \\ \Psi_2 \dfrac{\eta_x}{J} + \Psi_3 \dfrac{\eta_y}{J} = \dfrac{2\cos(\alpha)}{\gamma M_\infty^2 \bar{C}} \dfrac{dx}{d\xi} & on \;\; c \\ \Psi^T \delta G = 0 & on \;\; B \end{cases} \tag{56}$$

and Θ satisfies the following equation

$$\Theta \left(\cos(\alpha) \frac{dy}{d\xi} - \sin(\alpha) \frac{dx}{d\xi} \right) = \sin(\alpha) \frac{dy}{d\xi} \tag{57}$$

Equation (56) is the adjoint equation of Euler equations, (57) is additional algebraic adjoint equation which comes from the constraint on C_d, so the final form of variation of cost function becomes

$$\begin{aligned} \delta I = & - \int_D \Psi^T \left\{ \frac{\partial}{\partial \xi} \left[\delta(\frac{\xi_x}{J}) f + \delta(\frac{\xi_y}{J}) g \right] + \frac{\partial}{\partial \eta} \left[\delta(\frac{\eta_x}{J}) f + \delta(\frac{\eta_y}{J}) g \right] \right\} d\xi d\eta \\ & - \oint_c \Psi^T \left[\delta(\frac{\eta_x}{J}) \bar{f} + \delta(\frac{\eta_y}{J}) \bar{g} \right] d\xi + \frac{2\cos(\alpha)}{\gamma M_\infty^2 \bar{C}} \oint_c p \left(- \frac{d\delta x}{d\xi} + \Theta \frac{d\delta y}{d\xi} \right) d\xi \\ & + \frac{2\sin(\alpha)}{\gamma M_\infty^2 \bar{C}} \oint_c p \left(- \frac{d\delta y}{d\xi} - \Theta \frac{d\delta x}{d\xi} \right) d\xi \end{aligned} \tag{58}$$

Similiarly, we can do the further simplification to equation (58) using (32) (33), so that the above formulation can be rewritten similiarly to (34) (51), i.e.

$$\begin{aligned} \delta I = & - \int_D \Psi^T \delta R_{es}' d\xi d\eta - \oint_c \Psi^T \delta \bar{G}' d\xi + \frac{2\cos(\alpha)}{\gamma M_\infty^2 \bar{C}} \oint_c p \left(- \frac{d\delta x}{d\xi} + \Theta \frac{d\delta y}{d\xi} \right) d\xi \\ & + \frac{2\sin(\alpha)}{\gamma M_\infty^2 \bar{C}} \oint_c p \left(- \frac{d\delta y}{d\xi} - \Theta \frac{d\delta x}{d\xi} \right) d\xi \end{aligned} \tag{59}$$

Once the parameterization of airfoil shape is chosen, we can get a set of design variables(e.g. b_i), then calculate the gradient $\dot{g}$ of cost function with respect to design variables via (36). Here the design problem is maximization problem instead of minimization problem, so an improvement can be made with a shape change

$$\delta b = \lambda \dot{g} \tag{60}$$

that is to say: modify the airfoil along the gradient direction instead of minus gradient direction. The variation in the cost function then becomes

$$\delta I = \lambda \dot{g}^T \dot{g} \geq 0 \tag{61}$$

which means that this procedure will develop towards the maximization of lift.

Comparing the derivation in this section with that of in section 3, we find that if the cost function is different, the only change in the former development (section 3) relates to the boundary condition for the adjoint Euler equations and the computation of the final cost function variation, as can be seen in this section, but the remaining terms are kept the same. For the constrained optimization, the derivations indicate that it yields an additional adjoint equation with respect to the constraint, and with the corresponding revised boundary condition for the adjoint Euler equations and the final form of cost function variation, and the remaining terms are still kept the same. It turns out that control theory based optimization method is very convenient not only to treat a large number of design variables, but also constrained optimization.

5 Numerical Implementation

5.1 Numerical Procedure

The procedure can then be summaried as follows:

1. Solve the flow equations (11-14) and obtain ρ,u,v,p,e etc.

2. Solve the adjoint equations with the boundary conditions. For the unconstrained optimization, solve the adjoint Euler equation (29) with their boundary condition and obtain Ψ. For the drag minimization problem under the constraint of lift, solve the adjoint Euler equation (47) with their boundary conditions and the additional algebraic adjoint equation (48), for the lift constraint, obtain Ψ and Θ. For the lift maximization problem, solve adjoint equation (56) with their corresponding boundary conditions and the additional algebraic adjoint equation (57) for drag constraint, obtain Ψ and Θ.

3. Compute the metric variation $\delta(\xi_x/J)$ $\delta(\xi_y/J)$ $\delta(\eta_x/J)$ $\delta(\eta_y/J)$ and $d\delta s$ for each design variable at every point (ξ, η).

4. Evaluate δI for each design variable. For the unconstrained optimization use formulation (34), for the drag reduction problem under constraint of lift use (51), for the lift maximization problem use (59).

5. Introduce the gradient $\dot{g}$ into an optimization procedure.

6. Project the gradient into an allowable subspace that satisfies all the geometric constraints if necessary, obtain the projected gradient $\tilde{\dot{g}}$.

7. Update the shape based on the direction of descent.

8. Return to 1 until convergence.

5.2 Design Parameters

The airfoil geometry is modified by forming a linear combination of Hicks-Henne shape functions f_i as follows:

$$y^u_{new} = y^u_{initial} + \sum_{i=1}^{N} b^u_i f_i(x) \qquad y^l_{new} = y^l_{initial} + \sum_{i=1}^{N} b^l_i f_i(x)$$
$$f_i = \sin^{t2(i)}\left[\pi x^{t1(i)}\right] \qquad t1(i) = \frac{\ln(0.5)}{\ln(x_i)} \tag{62}$$

where b_i are the design variables, N the number of design variables, and x_i represents the peak location of f_i. Although these Hicks-Henne shape functions are not orthogonal, they have been widely used for aerodynamic shape optimization problems with successful results.

We define 14 Hicks-Henne design variables (coefficients of shape functions) for upper or lower surface distributions, normally $x(i)$ and $t2(i)$ are specified as indicated in table 1, and Fig.4 shows the Hicks-Henne functions corresponding to table 1.

Table 1: x(i) and t2(i) values for each shape function

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14
x(i)	.025	.05	.1	.15	.2	.3	.4	.5	.6	.7	.8	.85	.9	.95
t2(i)	2.	2.	2.	3.	3.	3.	3.	3.	3.	3.	3.	2.	2.	1.

5.3 Mesh Movement

During the optimization cycle, the geometry is modified and the computational mesh must be adapted. In this report, the **O**-type structure mesh is generated analytically by using the conformal mapping techniques[20], therefore after changing every design variables the mesh is regenerated analytically by solving two Laplace equations about (x, y) respect to (ξ, η) on the airfoil and kept the same at the outer boundary(details see [20]).

5.4 Euler Solver

The computational domain is discretized with an **O**-type grid, the governing equation and its boundary conditions cast in curvilinear coordinates are discretized with Jameson's finite-volume approach, in which second and fourth order artificial dissipation are blended to suppress the oscillations of the solution near the shock, which is then solved iteratively by using the implicit diagnolized factorization scheme.

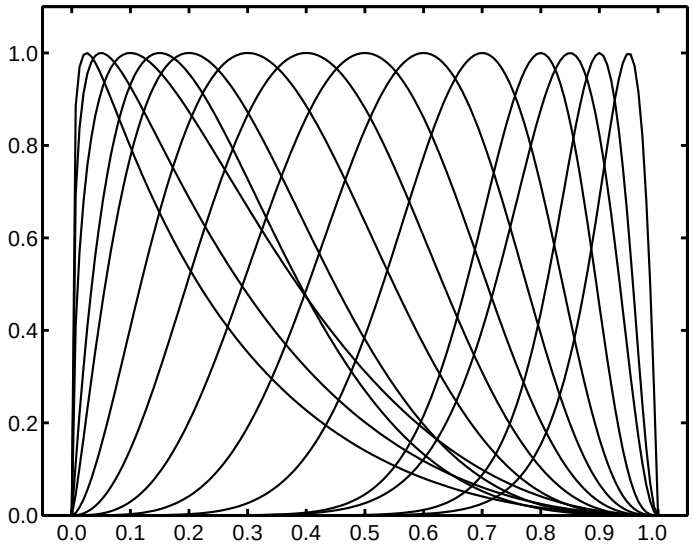

Figure 4: Fourteen Hicks-Henne shape functions used in this study

5.5 Adjoint Solver

Time derivatives are added to the adjoint Euler equations to obtain a time dependent system of equations, so the continuous adjoint Euler equation has a complexity similiar to that of the flow equation. Therefore adjoint Euler equations are discretized and solved in the same manner as the state equations, i.e. diagnolized-implicit factorization integration coupled with additional second and fourth order artificial dissipation terms; for a detailed discription and its boundary condition treatment see reference [20]. The additional adjoint equation arising from the constraint is linear, scalar and algebraic and it is solved using the common method.

5.6 Step Size Evaluation

At some initial b, any minimization process seeks to find a descent direction $\tilde{b}$ and a step size λ in which to change b such that

$$I(w + \lambda\tilde{w}, b + \lambda\tilde{b}) \leq I(w, b) \tag{63}$$

where $\lambda\tilde{w}$ is the corresponding change in w that satisfies the state equations. This process repeated several times until a minimum is reached.

The step size λ is determined by a line search. The objective of the line search is to find λ such that $\parallel \nabla_b I \left(w + \lambda\tilde{w}, b + \lambda\tilde{b}\right) \parallel^2$ is minimum, that is

$$\frac{\partial \parallel \nabla_b I \left(w + \lambda\tilde{w}, b + \lambda\tilde{b}\right) \parallel^2}{\partial\lambda} = 0 \tag{64}$$

By expanding in Taylor series, taking the derivative with respect to λ, and neglecting the $0(\lambda^2)$ terms, the step size becomes

$$\lambda = -\frac{\left[\nabla_b I(w,b)\right]^T \nabla_b^2 I(w,b)\tilde{b}}{\tilde{b}^T \left[\nabla_b^2 I(w,b)\right]^T \nabla_b^2 I(w,b)\tilde{b}} \tag{65}$$

where $\nabla_b^2 I$ is a symmetric matrix and is often referred to as the Hessian. Note that $\nabla_b^2 I$ includes the variation with respect to w. Computation of the Hessian is expensive, the cost is proportional to the number of design variables. However, $\nabla_b^2 I\tilde{b}$ can be evaluated relatively easily with finite defferences as follows

$$\nabla_b^2 I(w,b)\tilde{b} = \frac{\nabla_b I(w + \bar{\lambda}\tilde{w}, b + \bar{\lambda}\tilde{b}) - \nabla_b I(w,b)}{\bar{\lambda}} \tag{66}$$

where $\bar{\lambda}$ is a trial perturbation. To find the step size, the design variables are first perturbed with an arbitarily small $\bar{\lambda}$, and the new values of the state variables that satisfy the state equations are determined. Next, the new gradient $\nabla_b I(w + \bar{\lambda}\tilde{w}, b + \bar{\lambda}\tilde{b})$ is evaluated, followed by $\nabla_b^2 I(w,b)\tilde{b}$. Then the step size is determined with (65).

6 Nash Game

Nash made a start on the cooperative theory, and introduced the fundamental concept of equilibrium point. A Nash equilibrium is a steady state solution, such that no one player can improve his outcome by changing only his own strategy. The theory of Nash equilibrium has been used in study of political competition, in explanation of the distribution of tongue length, in revealing a sense in which the outcome of trading under a price system is stable in a economy that contains many agents.

In this work, we use the theory of Nash equilibrium to study the multi-objective optimum design in aerodynamics, especially in the designs in which mutli-criteria are in mutual conflict. Here, players are optimizers who are confronted with other optimizers, so at first we split the design variables into a number of subsets which number equal to the number of targets, each subset being associated with a player. The splitting of design variables depends on the physics of aerodynamics. Then allocate design targets to the players, each player optimize one's own criterion by modifying one's own subset (design variables) and keeping the other subsets unchanged with exchange of information with the others. A Nash equilibrium is reached when each player, constrained by the strategy of the others, cannot

improve further one's own target. let us assume that we have two design targets I_1 and I_2, under conflict oe not, these two targets are functions of the design variables X, so we split the design Variable X into two subsets x_1 and x_2, we further assume that the two targets are both minimization problems for convenience, and player 1 is responsible for I_1 , player 2 is responsible for I_2, so the design problem can be explained as follows:

$$\begin{array}{cc} Player1 & Player2 \\ min_{x_1} I_1(x_1, x_2) & min_{x_2} I_2(x_1, x_2) \end{array} \tag{67}$$

where x_1 is the free design variable of cost function I_1, x_2 is fixed for player 1 and comes from the result of player 2; similiarly x_2 is the free design variable of cost function I_2, x_1 is fixed for player 2 and comes from the result of player 1. If (x_1^*, x_2^*) is the equilibrium of this game, then we have:

$$\begin{array}{cc} Player1 & Player2 \\ x_1^* = min_{x_1} I_1(x_1, x_2^*) & x_2^* = min_{x_2} I_2(x_1, x_2^*) \end{array} \tag{68}$$

this usually happens when

$$\begin{array}{cc} I_{1\ x_1}(x_1^*, x_2^*) = 0 & I_{2\ x_2}(x_1^*, x_2^*) = 0 \\ I_{1\ x_1 x_1}(x_1^*, x_2^*) > 0 & I_{2\ x_2 x_2}(x_1^*, x_2^*) > 0 \end{array} \tag{69}$$

Therefore, if some solution satisfies (69), we say that design procedure reach their equilibrium and the process can be stopped.

In general, if $I_1 = I_2$, we call the above game a Virtual Game; if $I_1 \neq I_2$, we call the above game a Real Game. This tells us that Game theory can also be used as the optimization strategy to get the optimum solution for single point design problem; further in this report, this potential ability will be demonstrated by design experiments. Additionally, traditional mehtod is hard to be used in multi-objective designs when several targets are in conflict, above explanation tells us that Game theory provides an easy way to organize the balance between criteria in a multi-objective design; this will also be shown subsequently.

7 Distributed Computing

One important concern related with the multi-objective optimum shape design is the computational effort needed, because we have to analyse multi flow-field and adjoint field for each design condition. Distributed computing on multiple workstations is the underlying technology which makes an integrated design system possible by providing the computational resource neccessary to achieve acceptable execution times. In a symmetric Nash game, each

player attempts to optimize one's own target with exchange of symmetric information with the others. In that, each optimizer is independent of the others and so they can all be exented in parallel with one on each workstation. That is to say, in a Nash game each player is responsible for one target, then we distribute each design task into different machines, after several design cycles, each player get the better design variables corresponding to its target, then they exchange their information mutually. Repeat this process until no one player can improve his outcome by changing only one's own strategy, we say the equilibrium point of the game is reached, this equilibrium point is just the solution of the optimization problem and then design procedure stops.

8 Problems Under Study

8.1 Constrained Single Point Airfoil Drag Minimization

In this problem, the cost function is defined as

$$\begin{cases} minI = \dfrac{\Omega_1}{2}\oint_c (p-p_d)^2 ds + \Omega_2 C_d \\ subject\ \ to \qquad C_l = constant \end{cases} \tag{70}$$

where the weighting functions $\Omega_1 = 0.1$, $\Omega_2 = 2.9$. We choose the $RAE2822$ airfoil as the initial airfoil to reduce its pressure induced shock wave drag in transonic regime under the constraint of constant lift; the design conditions are $M_\infty = 0.73$, $\alpha = 2.0^0$, p_d is the initial pressure distribution of $RAE2822$ airfoil. So at first, we evaluate a transonic Euler flow at a Mach number equal to 0.73 and incidence equal to 2.0^0 on the $RAE2822$ airfoil, we calculate the pressure which is used as p_d, C_l is kept as the same as the initial one. We want to do a small modification to the initial airfoil and get a new airfoil whose drag is smaller than the initial one but the lift maintained.

8.2 Constrained Single Point Airfoil Lift Maximization

In this problem, the cost function is defined as

$$\begin{cases} maxI = C_l \\ subject\ \ to \qquad C_d = constant \end{cases} \tag{71}$$

we still choose the $RAE2822$ airfoil as the initial configuration to maximize its lift under the constraint of constant drag in subsonic regime; the design conditions are $M_\infty = 0.35$, $\alpha = 8.0^0$. We want to modify the inital airfoil and get a high lift configuration whose drag has not increased.

8.3 Two-Point Airfoil Drag Minimization

Here, we try to reduce the drag of the airfoil in two transonic regimes. The objective is to minimize simultaneously the following cost functions with their constraints:

$$\left\{\begin{array}{l} \left.\begin{array}{c} minI_1 = \dfrac{\Omega_1}{2}\oint_c (p-p_{d1})^2 ds + \Omega_2 C_{d1} \\ subject\ \ to \quad C_l = C_{l\ initial\ 1} \end{array}\right\} at\ \ M_\infty = 0.70, \alpha = 5.0^0 \\ \\ \left.\begin{array}{c} minI_2 = \dfrac{\Omega_1}{2}\oint_c (p-p_{d2})^2 ds + \Omega_2 C_{d2} \\ subject\ \ to \quad C_l = C_{l\ \ initial\ 2} \end{array}\right\} at\ \ M_\infty = 0.75, \alpha = 2.0^0 \end{array}\right. \tag{72}$$

Here, the initial airfoil is still $RAE2822$, the two weighting functions are still $\Omega_1 = 0.1$, $\Omega_2 = 2.9$. Firstly, we evaluate a transonic Euler flow at a $M_\infty = 0.70$, $\alpha = 5.0^0$ on $RAE2822$, we calculate the pressure p_{d1}, secondly, we evaluate the other transonic Euler flow at a $M_\infty = 0.75$, $\alpha = 2.0^0$, and we calculate the pressure p_{d2}. We try to modify the $RAE2822$ airfoil to reduce both drags simultaneously under these two design conditions and, if possible, maintain the lift to its initial value in both cases.

8.4 Two-Point Airfoil Lift Maximization/Drag Minimization

Here, we try to design a new airfoil which can produce higher lift in subsonic regime and lower drag in transonic regime than the inital values. The objective is to optimize simultaneously the following two cost functions with their constraints:

$$\left\{\begin{array}{l} \left.\begin{array}{c} maxI_1 = C_l \\ subject\ \ to \quad C_d = C_{d\ initial} \end{array}\right\} at\ \ M_\infty = 0.35, \alpha = 8.0^0 \\ \\ \left.\begin{array}{c} minI_2 = \dfrac{\Omega_1}{2}\oint_c (p-p_d)^2 ds + \Omega_2 C_d \\ subject\ \ to \quad C_l = C_{l\ \ initial} \end{array}\right\} at\ \ M_\infty = 0.75, \alpha = 2.0^0 \end{array}\right. \tag{73}$$

where, $\Omega_1 = 0.1$, $\Omega_2 = 2.9$, the initial airfoil is still $RAE2822$. At first, we compute a transonic Euler flow at a $M_\infty = 0.75$, $\alpha = 2.0^0$ over the RAE2822 airfoil, which pressure distribution is p_d. The design aims to making a proper modification on initial shape and getting a better lift performance in subsonic flow meanwhile its shock wave drag in transonic flow is as low as possible.

9 Results

In accordance with the above optimization problems mentioned in section 8 and design methods discussed in sections 3, 4. The following experiments have been performed to confirm the validality and effeciency of the present method.

9.1 Splitting Definition

According to the physics of aerodynamics, we perform the splitting of the airfoil as in the following three cases: Front/rear splitting, Front/Middle/Rear splitting, Alternating Splitting. These are illustrated in the following figures 5, 6, 7 respectively.

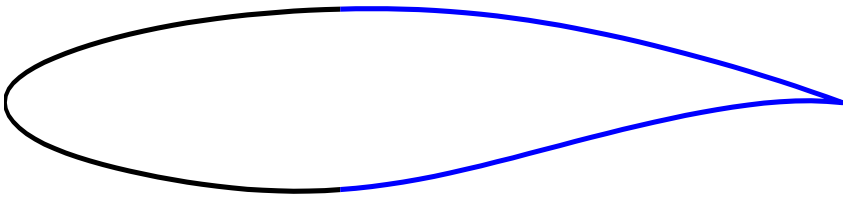

Figure 5: Diagram of Front/Rear splitting

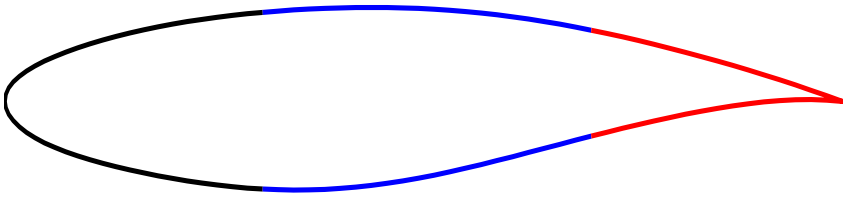

Figure 6: Diagram of Front/Middle/Rear splitting

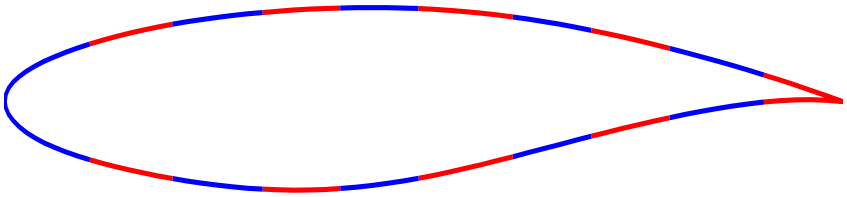

Figure 7: Diagram of Alternating splitting

9.2 Constrained Single Point Airfoil Drag Minimization via Global Optimization Method

The first demonstration is an application of the present method to treat the constrained optimization. The test case is described in section 8.1. The calculations were performed on a 128×38 mesh with 4864 cells, the mesh has a O-topology illustrated in Fig.2. The design iterations were carried out at $M_\infty = 0.73$, $\alpha = 2.0^0$, and the surface of airfoil is parameterized using 28 Hicks-Henne bell-shaped functions (see equation (63), table 1 and figure 4). In this test case, we attempt to minimize the drag induced by pressure in transonic flow without changing the lift coefficient. This design case is one of the most interesting problems since decreasing C_d without loss in C_l is most practical in Aerodynamic engineering. However, this case is also the most difficult problem, since a decrease in C_d usually comes at the expense of a decrease in C_l. Figure 8 shows the result of this type of design optimization, which indicates that the Euler design procedure produced a weaker shock. The numerical results presented show significant improvements in C_d, after 50 design cycles the resulting C_d decreased of 42.4% from 0.0092 to 0.0053, while the lift didn't change. Figure 9 shows the convergence and constraint histories of the design procedure which indicates the lift has been maintained while the drag decreased, it illustrates that the constrained optimization method developed in this paper is efficient and valid in the practical design case. Figure 10 shows the comparison of steady iso-mach lines on initial and final profiles, this also indicates that the final flow field has a weaker shock on the upper surface than the initial one.

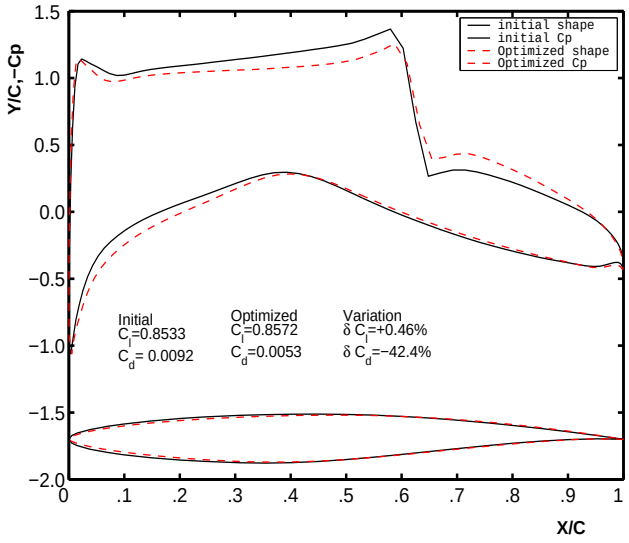

Figure 8: Comparison of final results with initial ones

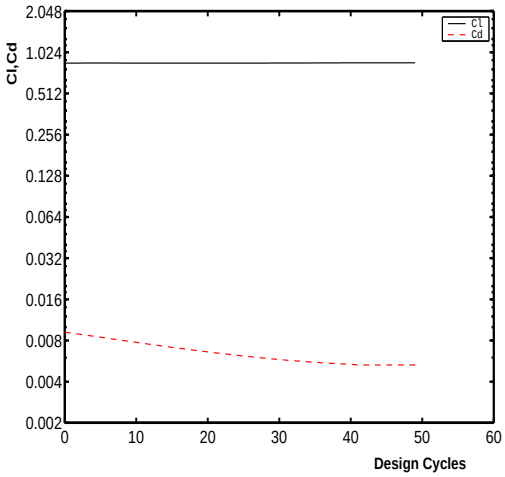

Figure 9: Drag and constraint histories

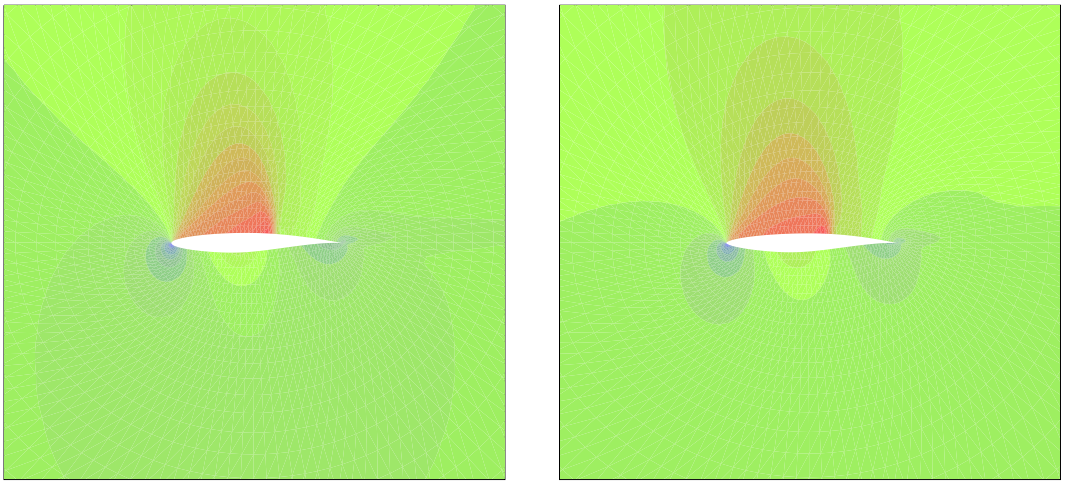

Figure 10: Steady iso-mach lines on initial and final profiles

9.3 Constrained Single Point Airfoil Lift Maximization via Global Optimization Method

In this test case which is described in section 8.2, we attempt to maximize the C_l of $RAE2822$ airfoil by modifying its shape using the same 28 Hicks-Henne design functions while constraining the coefficient of drag to be constant. Figure 11 shows the reults of this type of design optimization. The front portion of airfoil became thicker and the rear portion of airfoil has evolved towards the supercritical airfoil configuration. The numerical results presented show significant improvements in C_l, after 40 design cycles the resulting C_l increased of 20.9% from 1.1436 to 1.3826, while the drag didn't increase but reduced 1.43%. Figure 12 shows the convergence and constraint histories of the design procedure which indicates the drag has been maintained while lift increased. Figure 13 shows the comparison of steady iso-pressure lines on initial and final profiles, it can be seen that the slight difference between these two figures occurred at the trailling edge of the lower surface.

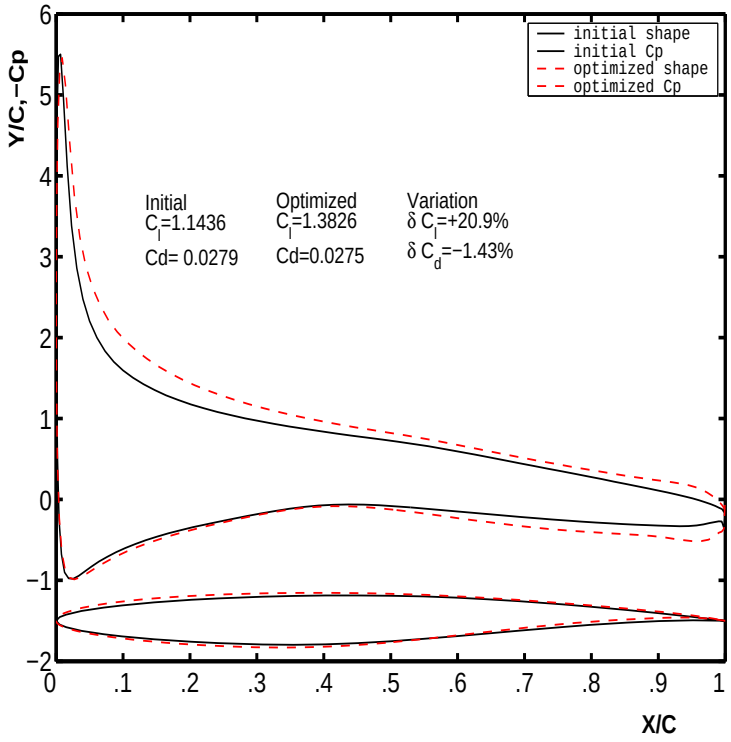

Figure 11: Comparison of final results with initial ones

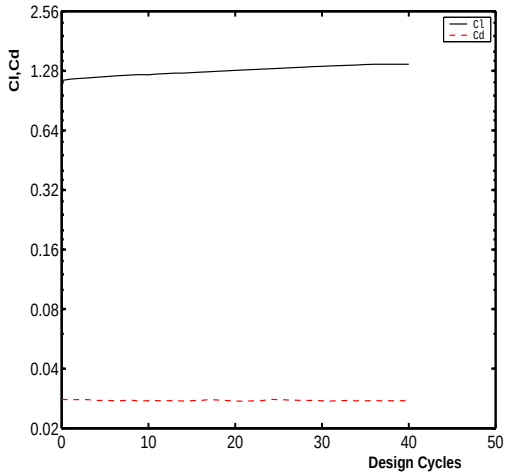

Figure 12: Lift and Constraint histories

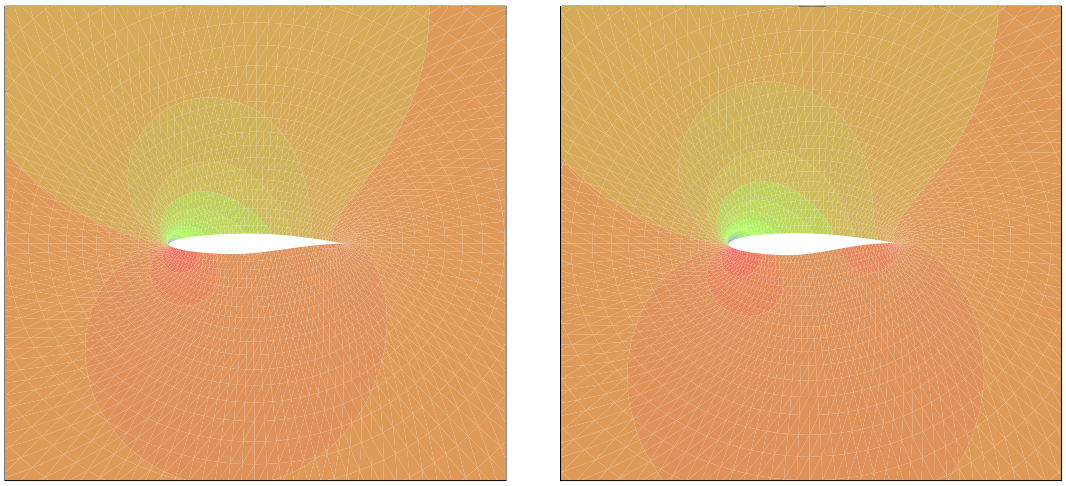

Figure 13: Steady iso-pressure lines on initial and final profiles

9.4 Single Point Airfoil Drag Minimization via Virtual Game Strategies

Here we redo the optimization problem defined in section 8.1, but using a Virtual Nash game strategy instead of a global optimization method, to show the potential ability of the game theory applied to optimum aerodynamic design. The design examples presented in this subsection were all carried out on a 128×38 mesh with 4864 cells, the three designs presented use the $RAE2822$ airfoil as starting point and the design were carried out at $M_\infty = 0.73$, $\alpha = 2.0^0$. The surface of the airfoil was parameterized using 28 Hicks–Henne bump functions, 14 of which are distributed along the upper surface of the airfoil, while the remaining 14 are placed in a similiar fashion along the lower surface (see table 1 and figure 4). Three design results are as follows:

9.4.1 Results of Front/Rear Splitting with Two Players

In this case, the splitting and optimization strategy are illustrated in figure 14, player 1 and player 2 optimize the same cost function (defined in (70)) but modify different portions of the airfoil. That is to say: split the design variables into two sets, the first set is responsible for the modification of front portion (20% of chord length from leading edge) of the airfoil, the second set is responsible for the modification of rear portion (80% of chord length from the trailing edge) of the airfoil, each player's task is distributed to different processor and modifies one's own design variables; meanwhile keep the opponent's design variables unchanged, then for each player, after 3 design iterations, each of them gets a better design and provide the other with one's own new design variables. Repeat this process untill no one player can improve its outcome by changing only one's own design variables. Figure 15 shows the reults of this type of design optimization, which indicates that the Euler design procedure produced a weaker shock. The numerical results presented show significant improvements in C_d, after 240 design iterations the resulting C_d decreased of 67.4% from 0.0092 to 0.0030, while the lift slightly increased of 3.33%. Figure 16 shows the drag histories of each player. Figure 17 shows the comparison of steady iso-mach lines on initial and final profiles, this also indicates that the final flow field has a weaker shock on the upper surface than the initial one.

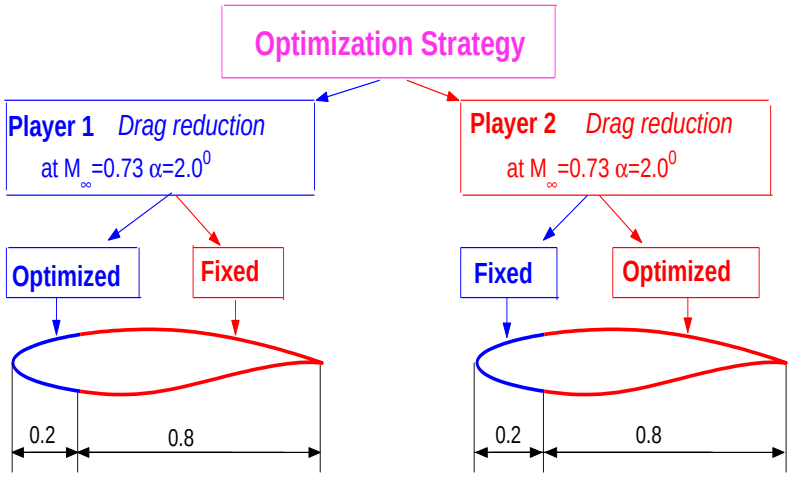

Figure 14: Splitting and optimization strategy
Exchange of information: Every 3—3 design iterations

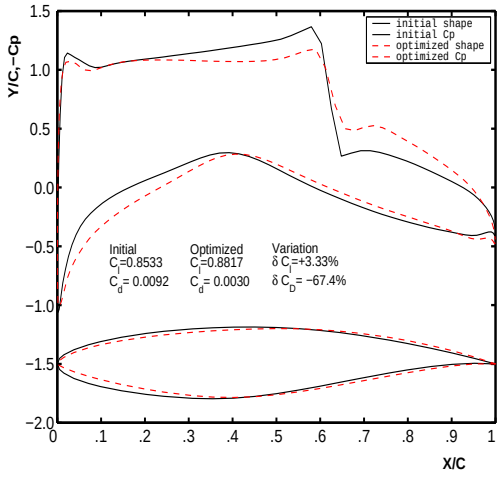

Figure 15: Comparison of final results with the initial ones

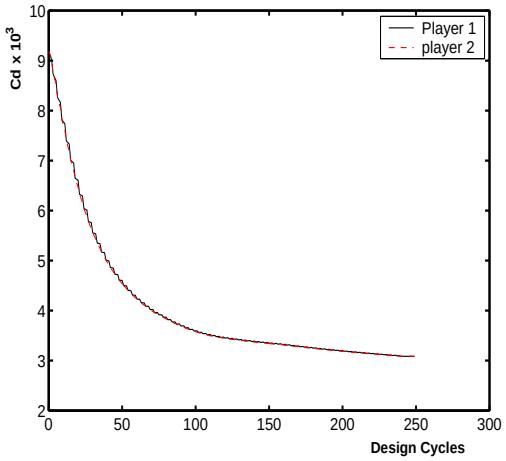

Figure 16: Drag histories of each player

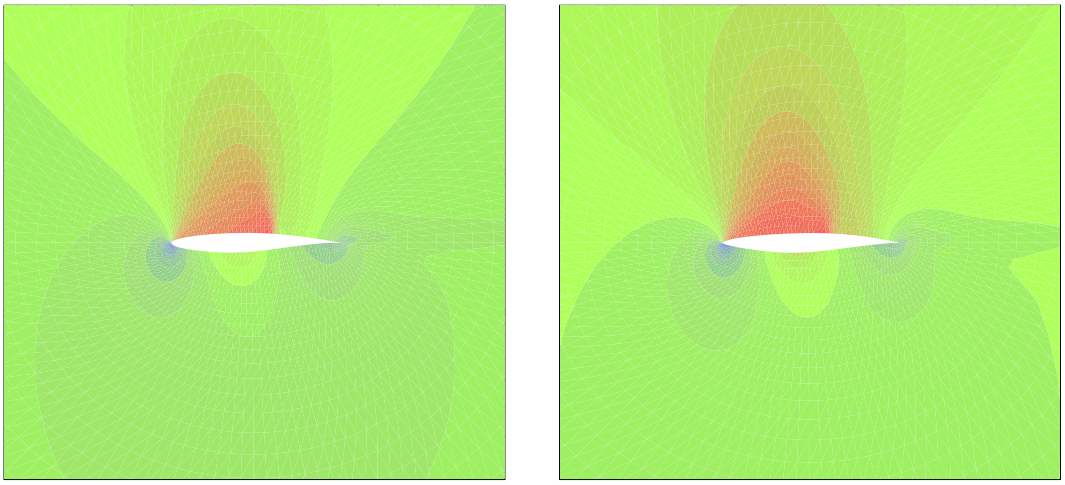

Figure 17: Steady iso-mach lines on initial and final profiles

9.4.2 Results of Front/Middle/Rear Splitting with Three Players

In this splitting, the optimization strategy is illustrated in figure 18, here we have three players, 1, 2 and 3. Each of them optimize the same cost function (defined in (70)) but modifies a different subset of the design variables. As can be seen from figure 18: the design variables are splitted into three sets, the first set is responsible for the modification of front portion of the airfoil (50% of chord length from leading edge), the second set is responsible for the modification of middle portion of the airfoil (25% of chord length next to first portion), the third set is responsible for the modification of rear portion of the airfoil (25% of chord length from trailing edge); each player's task is also distributed to different workstation for distributed computing and modifies one's own design variables meanwhile keep the opponent's design variables unchanged, then for each player, after 4–4–1 design iterations, each of them gets a better design and provides one's own new design variables to consist a new airfoil. Repeat this process until no one player can improve its outcome by changing only one's own design variables. Figure 19 shows the results of this type of design optimization, which indicates that the Euler design procedure really produced a weaker shock. The numerical results presented show significant improvements in C_d, after 124 design iterations the resulting C_d decreased of 66.3% from 0.0092 to 0.0031, while the lift slightly increased of 3.16%. Figure 20 shows the drag histories of the design procedure. Figure 21 shows the comparison of steady iso-mach lines on initial and final profiles, this also indicates that the final flow field has a weaker shock on the upper surface than the initial one.

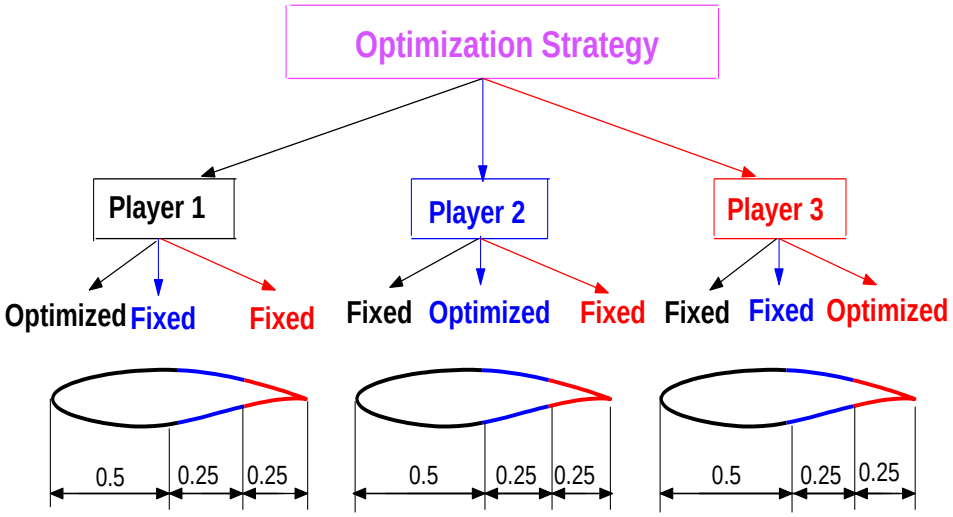

Figure 18: Splitting and optimization strategy
Exchange of information: Every 4—4—1 design iterations

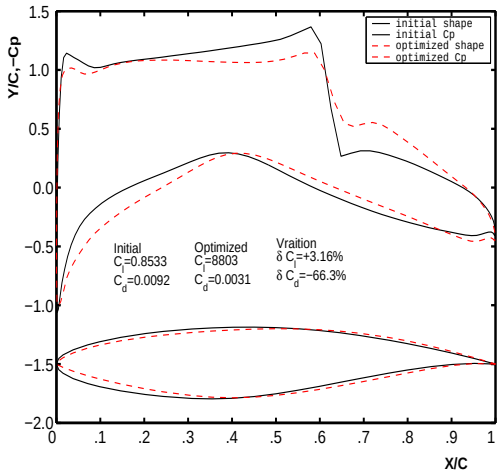

Figure 19: Comparison of final results with the initial ones

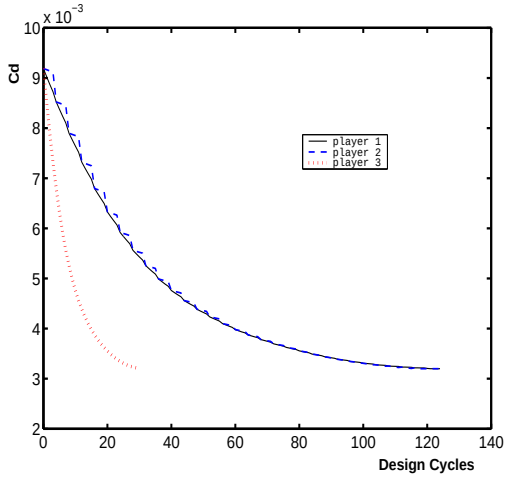

Figure 20: Drag histories of each player

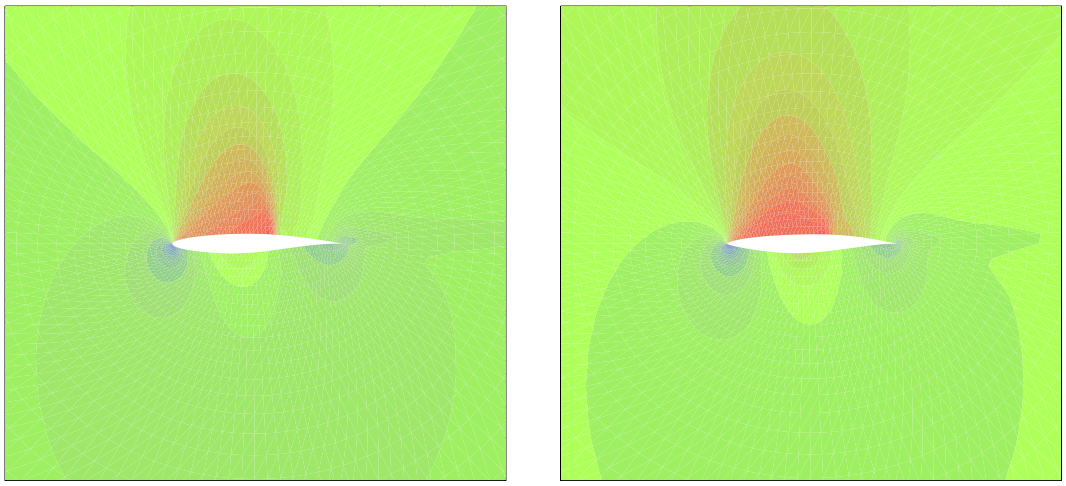

Figure 21: Steady iso-Mach lines on initial and final profiles

9.4.3 Results of Alternating Splitting with Two Players

In this case, the splitting and optimization strategy are illustrated in figure 22, player 1 and player 2 still optimize the same cost function (defined in (70)) but modify different portions of the design variables. i.e. split the design variables into two sets, the first set is responsible for the design variables of odd index, the second set is responsible for the design variables of even index, each player's task is distributed to a different workstation and modifies one's own design variable meanwhile keep his opponent's design variables unchanged, then for each player, after 3 design iterations, each of them gets a better design and provides one's own new design variables to consist a new airfoil. Repeat this process until no one player can improve its outcome by changing only one's own design variables. Figure 23 shows the reults of this type of design optimization, which indicates that the Euler design procedure produced a weaker shock. The numerical results presented show significant improvements in C_d, after 248 design iterations the resulting C_d decreased of 66.3% from 0.0092 to 0.0031, while the lift slightly increased of 3.81%. Figure 24 also shows the drag histories of the design procedure. Figure 25 shows the comparison of steady iso-mach lines on initial and final profiles, this also indicates that the final flow field has a weaker shock on the upper surface than the initial one.

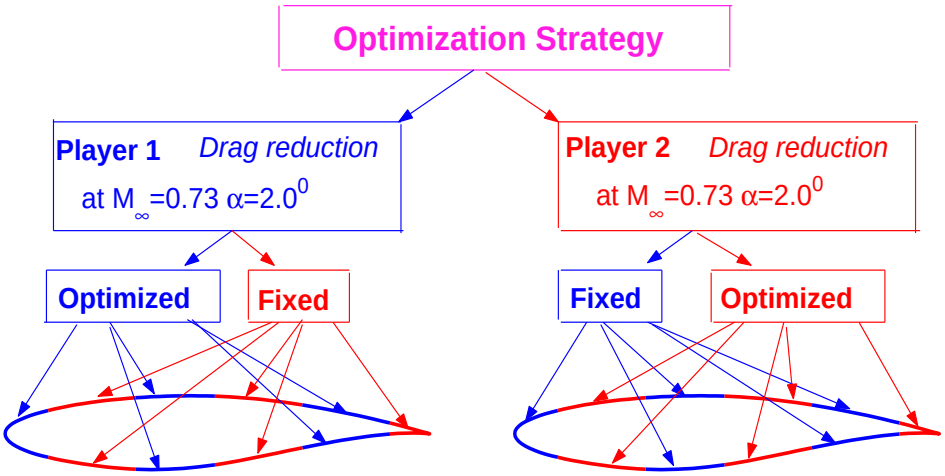

Figure 22: Splitting and optimization strategy
Exchange of information: Every 3—3 design iterations

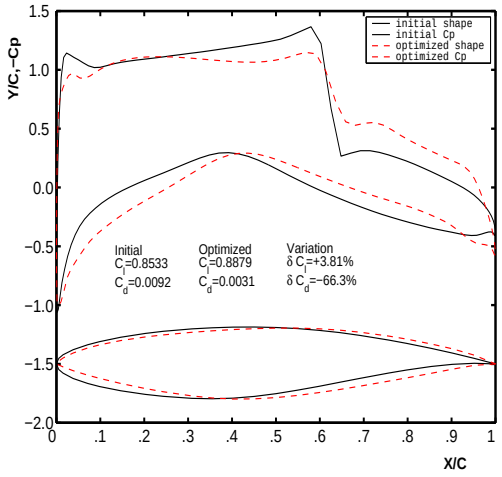

Figure 23: Comparison of final results with initial ones

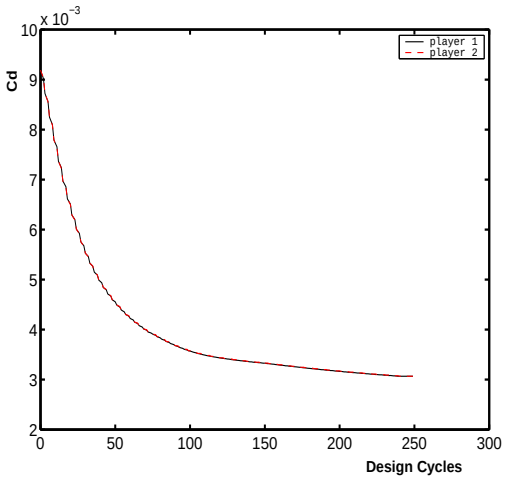

Figure 24: Drag histories for each player

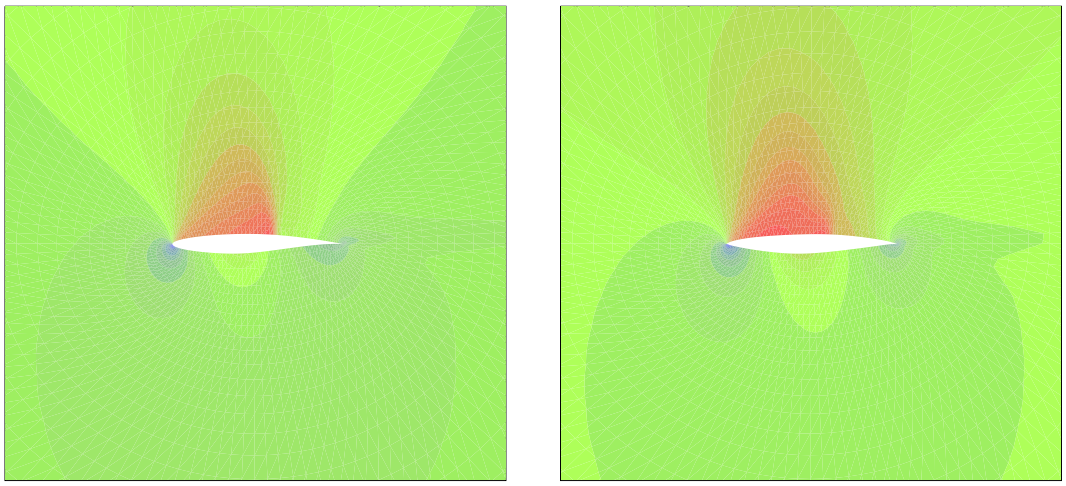

Figure 25: Steady iso-Mach lines on initial and final profiles

Comparing the three results presented here, we can draw the following conclusion: for the single point optimization, we can get almost the same optimization result via virtual Nash game no matter which kind of splitting and design strategy we use, but the computational speed varies from case to case, this is illustrated in figure 26. It shows that computation can be speeded up by splitting the design task as multi sub-tasks and distributing each sub-task to a different processor; moreover, the calculation is more efficient when the number of the sub-task increases. This is an important advantage of the game theory in parallel computing.

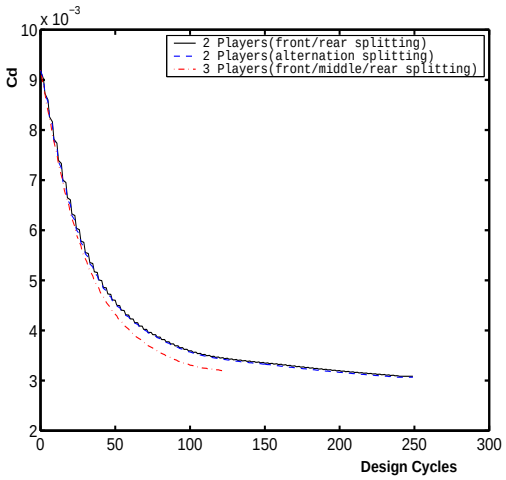

Figure 26: Comparison of drag histories with different number of players and splitting

9.5 Two-Point Airfoil Drag Minimization via Real Game Strategies

Here we perform the multi-objective optimization design problem described in section 8.3 using real Nash game strategies, to show the flexiblity and efficiency of game theory when applied to multi-objective aerodynamic optimization design. The design examples presented in this subsection were all carried out on a 128×38 mesh with 4864 cells, the two designs presented use the $RAE2822$ airfoil as starting point and the design were carried out at two transonic design points: $M_\infty = 0.70$, $\alpha = 5.0^0$, and $M_\infty = 0.75$, $\alpha = 2.0^0$. The surface of the airfoil was parameterized using 28 Hicks–Henne bump functions, 14 of which are distributed along the upper surface of the airfoil (see table 1 and figure 4), while the remaining 14 placed in a similiar fashion along the lower surface. Two design results are as follows:

9.5.1 Results of Front/Rear Splitting with Two Players

In this design case, the splitting and optimization strategy are illustrated in figure 27, player 1 and player 2 minimize simultaneously the drag of airfoil in two different transonic regimes. Each player is responsible for one design point by modifying a different portion of the design variables. Here, the design variables are still splitted into two sets, the first set is responsible for the modification of the front portion of the airfoil (20% of chord length from leading edge), the second set is responsible for the modification of rear portion of the airfoil (80% of chord length from the trailing edge); each player's task is distributed to a different workstation and modifies one's own design variables meanwhile keeping the opponent's design variables unchanged, then for each player, after 2 design iterations, each of them gets a better design and provides one's own new design variables to consist a new airfoil. Repeat this process until no one player can improve its outcome by changing only one's own design variables. Figure 28 shows the reults of this type of design optimization, which indicates that the Euler design procedure produced the weaker shock for both design points. The numerical results presented show significant improvements in C_d for each design point, after 140 design iterations the resulting C_d of player 1 decreased of 26.0% from 0.0501 to 0.0371, while the lift slightly decreased of 7.51%, and the resulting C_d of player 2 decreased of 80.3% from 0.0223 to 0.0044, while the lift slightly increased of 1.03%. Figure 29 shows the both drag histories of the design procedure, which indicates that the Nash equilibrium is really reached because both drags didn't change any more. Figure 30 shows the comparison of steady iso-mach lines on initial and final profiles for the player 1, Figure 31 shows the comparison of steady iso-mach lines on initial and final profiles for the player 2, these also indicate that both final flow fields have a weaker shock on the upper surface than the initial ones.

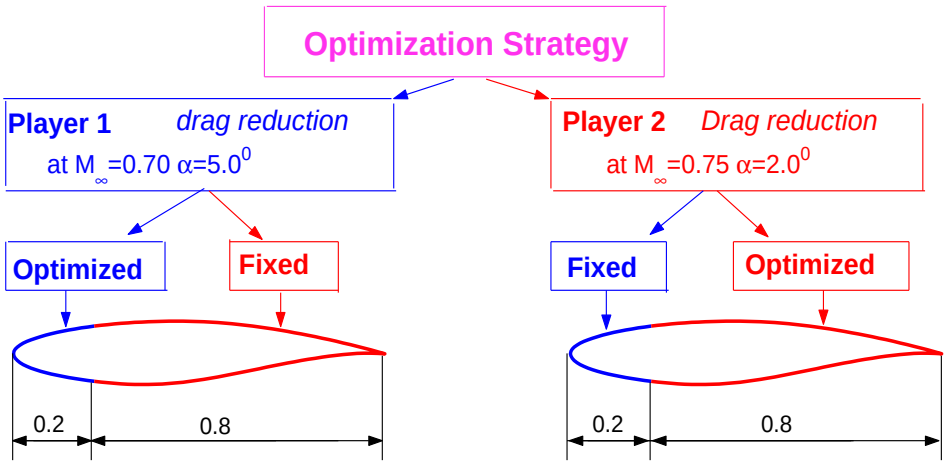

Figure 27: Splitting and optimization strategy
Exchange of information: Every 2—2 design iterations

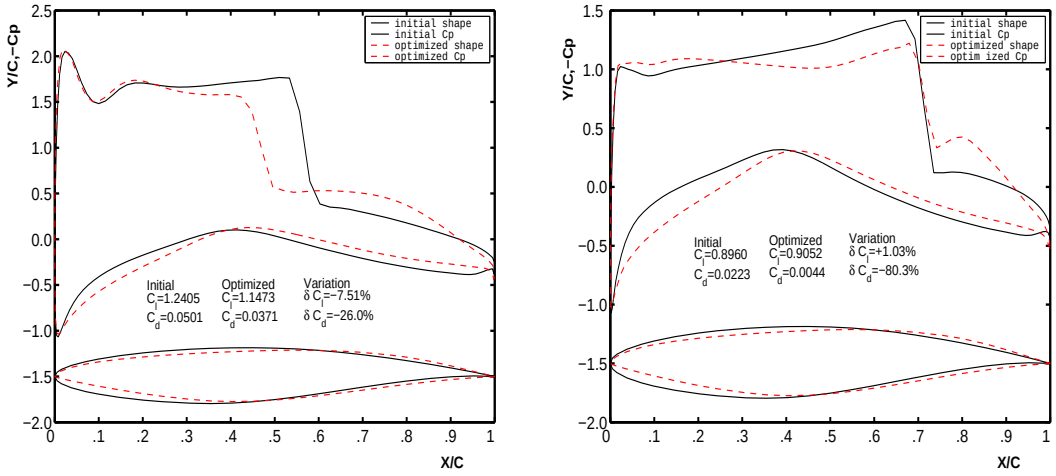

Figure 28: Comparison of final results with the initial ones for two design points

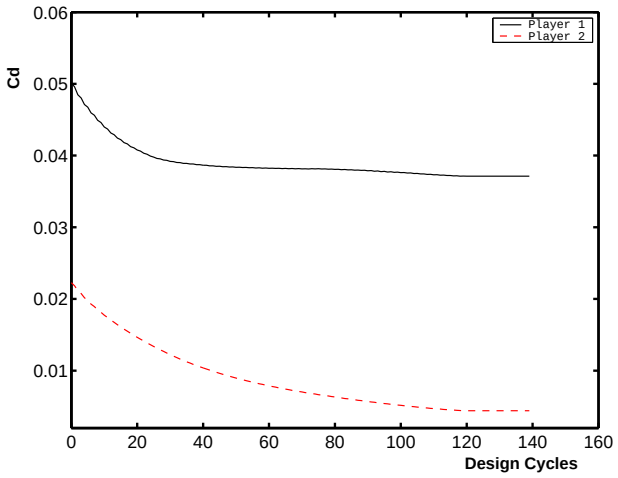

Figure 29: Convergence histories of the design procedure

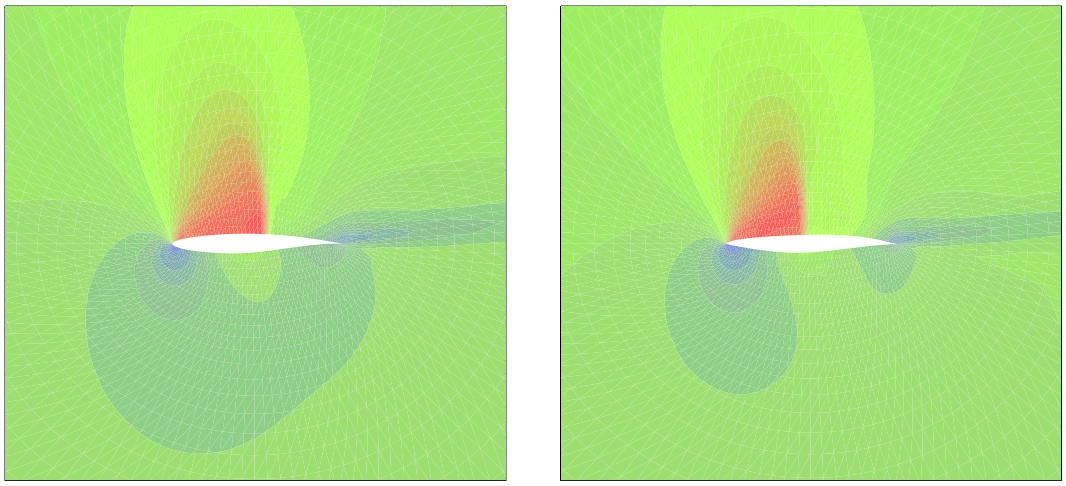

Figure 30: Steady iso-Mach lines on initial and final profiles for the first player

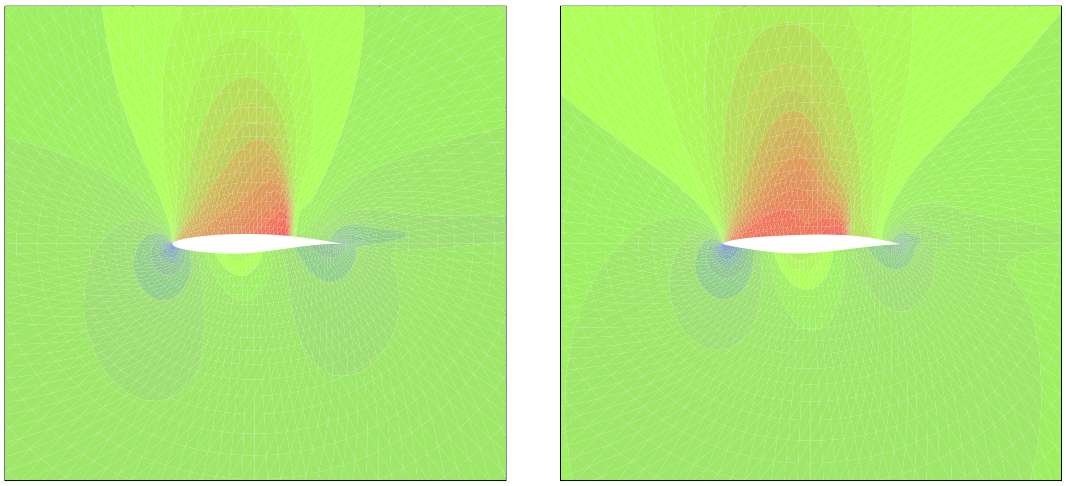

Figure 31: Steady iso-Mach lines on initial and final profiles for the second player

9.5.2 Results of Alternating Splitting with Two Players

Here, the splitting and optimization strategy are illustrated in figure 32, two players minimize simultaneously the airfoil drag in two different transonic regimes. Each player is responsible for one design point by modifying a different portion of the design variables. That is to say: split the design variables into two sets, the first set is responsible for the design variables of odd index, the second set is responsible for the design variables of even index (see table 1 and figure 4), each player's task is distributed to a different workstation and modifies one's own design variables meanwhile keeping the opponent's design variables unchanged, then for each player, after 2 design iterations, each of them gets a better design and provides one's own new design variables to consist a new airfoil. Repeat this process until no one player can improve its outcome by changing only its own design variables. Figure 33 shows the reults of this type of design optimization, which indicates that the Euler design procedure produced the weaker shock for both design points. The numerical results presented show significant improvements in C_d for each design point, after 140 design iterations the resulting C_d of player 1 decreased of 32.9% from 0.0501 to 0.0336, while the lift slightly decreased of 4.33%, and the resulting C_d of player 2 decreased of 65.6% from 0.0223 to 0.0077, while the lift slightly decreased of 4.07%. Figure 34 shows both drags histories of the design procedure, which indicate that the Nash equilibrium is really reached because both drag didn't change any more. Figure 35 shows the comparison of steady iso-mach lines on initial and final profiles for the player 1, Figure 36 shows the comparison of steady iso-mach lines on initial and final profiles for the player 2, these also indicate that both final flow fields present a weaker shock on the upper surface than the initial ones.

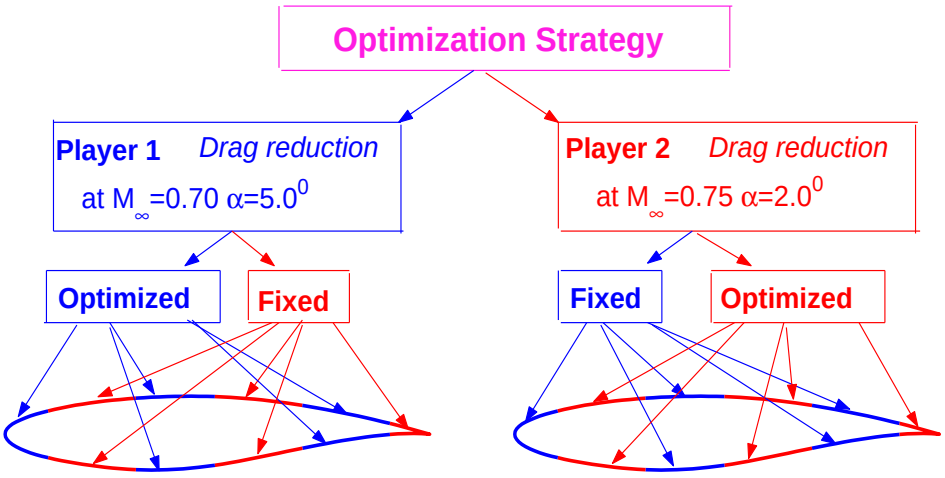

Figure 32: Splitting and optimization strategy
Exchange of information: Every 2—2 design iterations

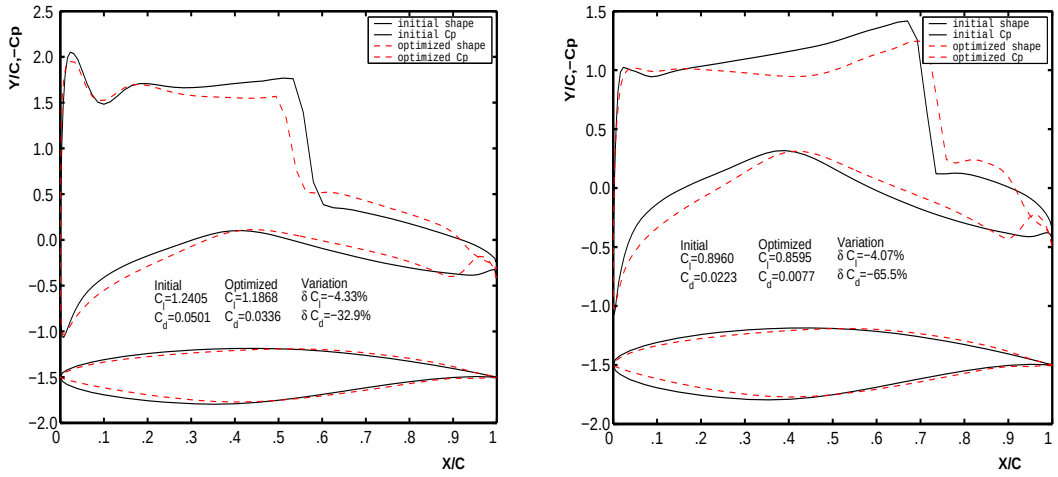

Figure 33: Comparison of final results with the initial ones for two design points

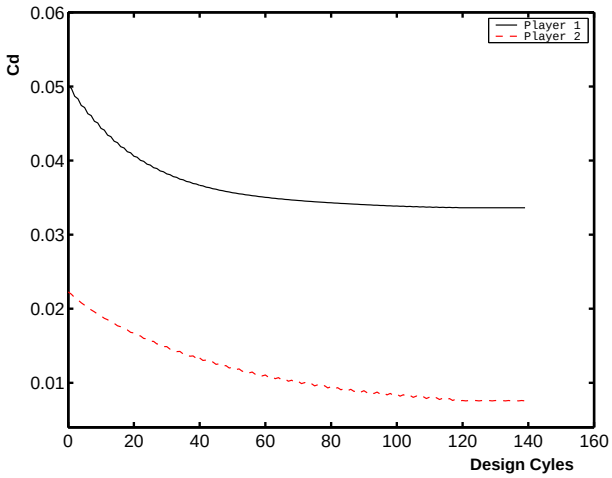

Figure 34: Convergence histories of the design procedure of two Players

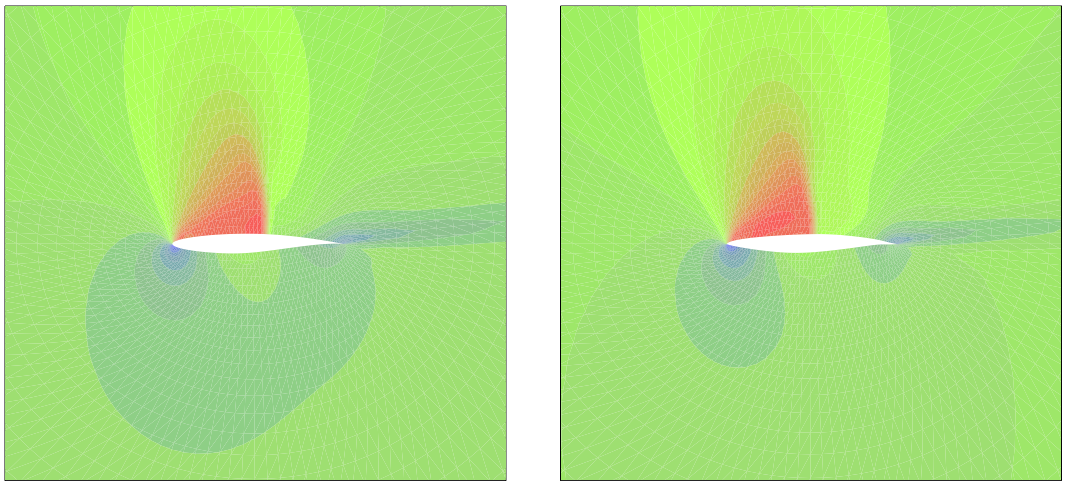

Figure 35: Steady iso-Mach lines on initial and final profiles of the first player

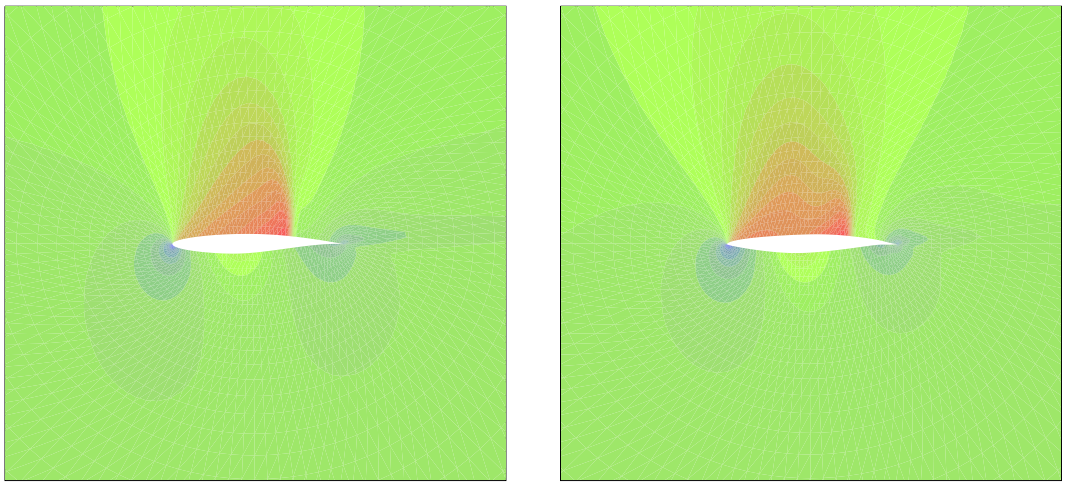

Figure 36: Steady iso-Mach lines on initial and final profiles of the second player

9.6 Two-Point Airfoil Lift Maximization/Drag Minimization Design via Real Game Strategy

This example is the core of this work. We will treat the multi-objective optimum design problem to show further the flexiblity and efficiency of game theory applied to multi-objective aerodynamic design, in which two targets are in conflict. For this kind of design problem, the traditional multi-objective optimization design method exposes its inherent shortcoming, because it is difficult to organize the design problem under conflict. But it can be easily treated by game theory, and in such treatment relies the greatest potential of the approach. The test case is defined in section 8.4. The design examples presented in this subsection were all carried out on a 128×38 mesh with 4864 cells, the design starts from the $RAE2822$ airfoil and the design were carried out at two design points: $M_\infty = 0.35, \alpha = 8.0^0$ for the first player to maximize lift, and $M_\infty = 0.75$, $\alpha = 2.0^0$ for the second player to reduce the transonic shock wave drag. The surface of the airfoil was parameterized using 28 Hicks–Henne bump functions, 14 of which distributed along the upper surface of the airfoil (see table 1 and figure 4), while the remaining 14 placed in a similiar fashion along the lower surface. The splitting and optimization strategy are illustrated in figure 37, player 1 and player 2 maximize lift/minimize drag of airfoil simultaneously in two different design points. Each player is responsible for one design point by modifying the different portion of design variables, that is to say: split the design variables into two sets, the first set is responsible for the modification of front portion of the airfoil (10% of chord length from leading edge) according the lift maximization criterion, the second set is responsible for the modification of rear portion (90% of chord length from the trailing edge) of the airfoil according the drag minimization criterion, each player's task is distributed to a different workstation and modifies one's own design variables meanwhile keeping the opponent's design variables unchanged, then for each player, after 3 design iterations, each of them gets a better design and provides one's own new design variables to consist a new airfoil. Repeat this process until no one player can improve its outcome by changing only its own design variables. Figure 38 shows the results of this design optimization, which indicates that the design procedure produced higher lift for player 1 in subsonic flow and lower drag for player 2 in transonic flow. The numerical results presented show remarkable improvements in C_l for subsonic design point, after 81 design iterations the resulting C_l of player 1 increased of 6.38% from 1.1436 to 1.2166, meanwhile the drag decreased of 41.9%, and significant improvement in C_d for transonic design point, the resulting C_d of player 2 decreased of 34.5% from 0.0223 to 0.0146, while the lift slightly decreased of 0.54%. Figure 39 show the variation procedure of lift and drag for each design point, which indicates that the Nash equilibrium is really reached, because all of them didn't change any more with the increase of design cycles numbers. Figure 40 shows the comparison of steady iso-pressure lines on initial and final profiles for the player 1, Figure 41 shows the comparison of steady iso-mach lines on initial and final profiles for the player 2, these also indicate that the final flow fields are superior to the initial ones concerning each design point.

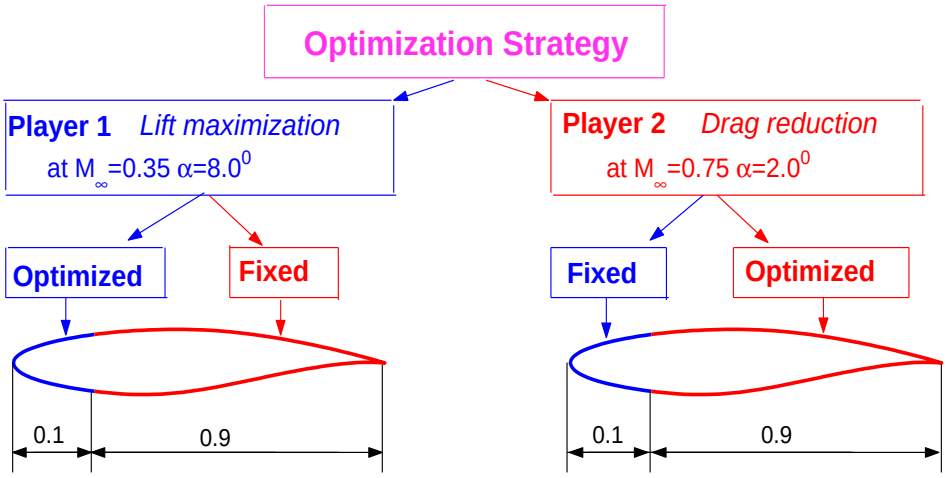

Figure 37: Splitting and optimization strategy
Exchange of information: Every 3—3 design iterations

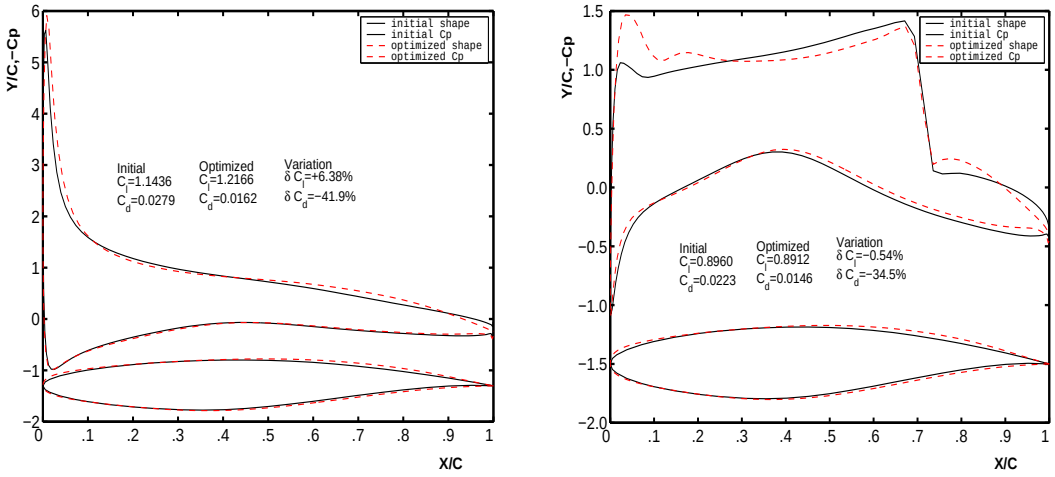

Figure 38: Comparison of final results with the initial ones for two design points

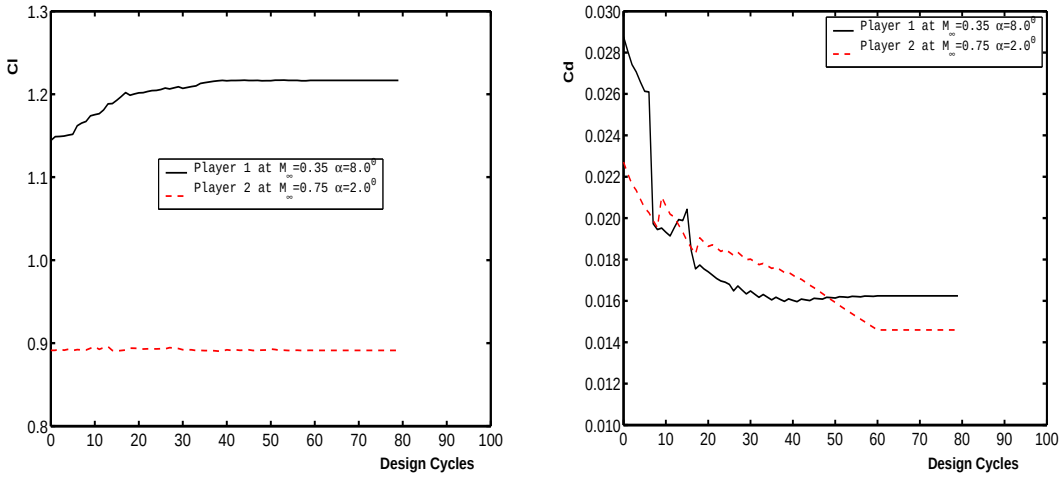

Figure 39: Convergence histories of the design procedure

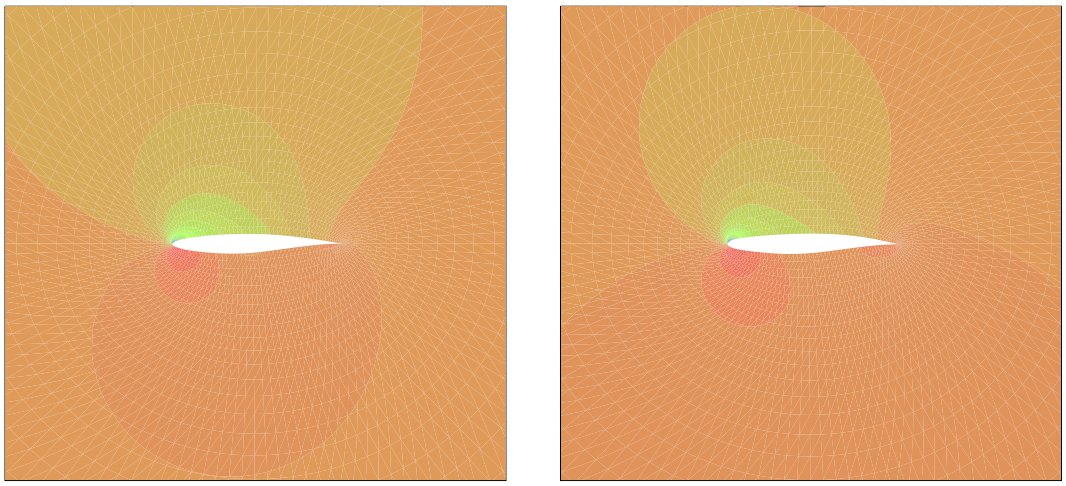

Figure 40: Steady iso-pressure lines on the initial and final profiles for the first player

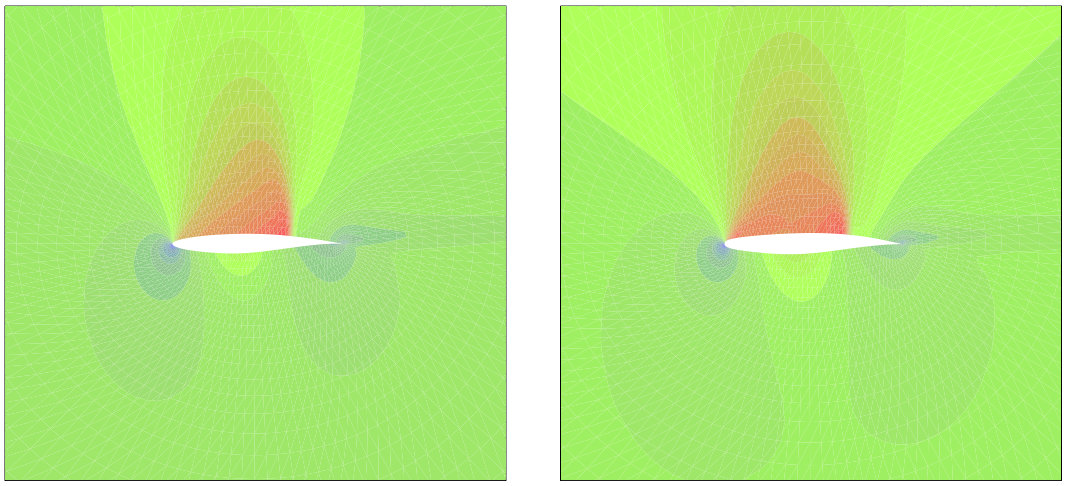

Figure 41: Steady iso-mach lines on the initial and final profiles for the second player

For this design case, if we exchange player 1 and player 2, that is to say: player 1 designs the rear portion of airfoil to maximize the lift in subsonic regime; player 2 designs the front portion(leading edge) of airfoil to minimize the pressure induced drag in transonic regime. We say that this kind of splitting and optimization strategy are non-physical, because the rear portion of airfoil shape is very sensitive to drag property in transonic regime, but we don't design this portion to reduce drag, meanwhile leading edge shape is also very important to subsonic aerodynamic performance of airfoil, but we don't design leading edge according the subsonic criterion. So we say this design strategy is unreasonable. As a result we have verified that the Nash Game does not work, players can not reach the equilibrium point. Actually, the experiment indicates that the drag in transonic regime increases and lift in subsonic regime reduces.

10 Conclusion

In this work, we have introduced a novel approach combining the Adjoint-Equation technique with a formulation derived from Game Theory to treat multi-point airfoil optimization problems. Airfoil shapes are optimized according to various aerodynamic criteria. In a symmetric Nash game, the notion of 'players' is introduced and each player attempts to optimize one's own target with symmetric exchange of information with the others. A Nash equilibrium is reached when each player, constrained by the strategy of the others, cannot improve further one's own target. Here, the Adjoint-Equation technique provides a cheap optimizer, and the Nash Game formulation has two merits : it provides a simple algorithm to organize the balance between criteria under conflict in a multi-disciplinary or multi-

point design, and it allows a very efficient computation by distributing tasks in a parallel environment.

For the single point optimization, the computation can be speeded up by using a Virtual game (i.e. split the design task into several sub-tasks to be distributed to different processors). Moreover, the calculation is more efficient when the number of the sub-tasks increases. This is an important advantage of the game strategy formulation for parallel computing.

We have applied concepts of control theory to treat the constraint explicitly in the optimizer by introducing a scalar adjoint variable; this yields to an additional linear scalar algebraic adjoint equation which is easy to solve. Compared with the unconstrained case, the derivations indicate that the only changes are : the additional adjoint equation to solve, the boundary conditions to the adjoint Euler equations and the final form of cost function variation to be modified. It turns out that the control theory based optimization method is very suitable not only for the designs involving a large number of design variables, but also for treating constrained optimization.

Our treatment of multi-point or multi-disciplinary optimization relies crucially on a split of territories (made here *a priori*) and the algorithm employed to organize the information exchange. This split should be devised to account on the flow physics in a sensible way. This is necessary for the Nash game algorithm to converge, and the optimization to succeed.

References

[1] J.L. Lions, *Contrôle Optimal des Systèmes Gouvernés par des Equations aux Dérivées Partielles*, Gauthier-Villars, Paris, 1969.

[2] M.J. Lighthill, *A new method of two dimensional aerodynamic design*, ARC, Rand M 2112, 1945

[3] J.L. Steger, J.M. Klineberg, *A finite difference method for transonic airfoil design*, AIAA J., Vol.11, No.5, PP628-635, 1973.

[4] G. Volpe, R.E. Melnik, *The design of transonic airfoil by a well posed inverse method*, Int. J. Numerical Methods in Engineering, Vol.22, PP341-361, 1986.

[5] G. Kuruvila, S. Táasan, M.D. Salas, *Airfoil design and optimization by the one-shot method*, AGARD-FDP-VKI Special course on optimum design methods for aerodynamic, AGRD Report 803, November, 1994.

[6] G.B. Cosentine, T.L. Holst, *Numerical optimization design of advanced transonic wing configuration*, NASA TM-85959, 1984

[7] K.D. Lee, S. Eyi, *Transonic airfoil design by constrained optimization*, AIAA 91-3287, AIAA 9th Applied aerodynamic conference, September 1991.

[8] G.A. Wrenn, *An indirect method for numerical optimization using the Kreisselmeier Steinhauser function*, NASA CR-4220, 1989

[9] J. Périaux, M. Sefrioui, B. Stoufflet, B. Mantel, E. Laporte, *Robust genetic algorithms for optimization problems in aerodynamic design*, In M. Galais, G. Winter, J. Périaux and P. Gueslta editors, Genetic Algorithm in Engineering and Computer Science, John Wiley &Sons, PP371-396, 1995.

[10] P. Hajela, J. Lee, *Genetic algorithms in multidisciplinary rotor blade design*, Proceedings of the 36th AIAA/ASME/ASCE/AHS/ASC/SDM conference, New Orleans, Louisiana, AIAA 95-1144, PP2187-2197, 1995.

[11] A. Jameson, *Aerodynami design via control theory*, Journal of Scientific Computing, Vo.3, No.3, 1988.

[12] J. Reuther, A. Jameson, *Control theory based airfoil design using the euler equations*, AIAA 94-4272 Cp, 1994

[13] O. Baysal, G.W. Burgreen, *Aerodynamic shape optimization using preconditioned conjugate gradient methods*, In 11th AIAA Computational fluid dynamic conference, Orlando, Florida, July, 1993.

[14] S. Hiernaux, J.A. Essers, *Aerodynamic optimization using Navier-Stokes equations and optimal control theory*, AIAA 99-3297, 1999.

[15] G.W. Burgreen, O. Baysal, *Three-dimensional aerodynamic shape optimization using discrete sensitivity anlysis*, AIAA J., Vol.34, No.9, 1996.

[16] W.K. Anderson, V. Venkatarishman, *Aerodynamic shape optimization on unstructured grids with a continuous adjoint formulation*, 35th Aerospace Science Meeting and Exihibit, Reno, January, 1997.

[17] J. Reuther, A. Jameson, J.J. Alonso, M.J. Rimlinger, D. Saunders, *Constrained multipoint aerodynamic shape optimization using an adjoint formualtion and parallel computers*, Part1, Journal of Aircraft, Vol.36, No.1, PP51-60, 1999.

[18] J.K. Hyoung, D.S. Shigern, O. Baysal, K. Nakahashi, *Aerodynamic optimization of supersonic transport wing using unstructured adjoint method*, AIAA J., Vol.39, No.6, PP1011-1020, 2001.

[19] N. Marco, J.A. Désidéri, S. Lanteri, *Multi-objective optimization in CFD by genetic algorithms*, INRIA report No.3686, 1999

[20] Z.L. Tang, *The research on optimum aerodynamic design using CFD and control theory*, PhD thesis, NUAA, P.R. China, 2000.

[21] J.F. Nash, *Non-cooperative games*, Annals of Mathematics, 54, 289, 1951.

[22] R.M. Hicks, P.A. Henne, *Wing design by numerical optimization*, Journal of aircraft, Vol.15, No.7, PP407-412, 1978.

[23] O. Pironneau, *Optimal shape design for elliptic systems*, Spring-Verlag, New York, 1984.

[24] O. Pironneau, *Optimal shape design for aerodynamics*, AGARD report 803, 1994.

[25] O. Baysal, M.E. Eleshaky, *Aerodynamic design optimization using sensitivity analysis and computational fluid dynamics*, AIAA Journal Vol.30, No.3, PP718-725, 1992.

[26] H. Cabuk, C.H. Shung, V. Modi, *Adjoint operator approach to shape design for internal incompressible flow*, In G.S. Dulikravish, editor, proceedings of the 3rd international conference on inverse design and optimization in Engineering science, PP391-404, 1991.

[27] Z.L. TANG, J.A. DESIDERI, J. PÉRIAUX,*Distributed optimization using virtual and real game strategies for Aerodynamic design*, WEHSFF 2002 Conference, Marseilles, 22-26 April, 2002

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38330 Montbonnot-St-Martin (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399